

Article

Using Active Methodologies for Teaching and Learning of Exploratory Test Design and Execution

Igor Ernesto Ferreira Costa *, Sandro Ronaldo Bezerra Oliveira , Isaac Souza Elgrably, Anderson dos Santos Guerra, Elziane Monteiro Soares and Iuri Victor Ferreira Costa

Graduate Program in Computer Science (PPGCC), Federal University of Pará (UFPA),
Belém 66075-110, Pará, Brazil

* Correspondence: igor.ef.costa@icen.ufpa.br

Abstract: Background: Exploratory testing has been used in the industry mainly to meet the needs of agile testing processes. In an agile scenario, this testing approach is important to facilitate changes constantly, which helps agile projects successfully. One of the ways to achieve success more easily is if the test process follows guides that provide structured activities. Problem Analysis: In this context, some studies have identified that software engineering does not understand how to apply exploratory testing approaches in a systematic way, involving both design and execution activities. After a review of the literature, it has been noticed that professionals performing exploratory testing focus only on execution activities. In an academic scenario, the literature review has evidenced that content about exploratory testing is not usually taught, and it is important that the professional acquires enough basic skills to apply it through structured activities. Purpose: In this context, this article presents an analysis and discussion of the results obtained in an experiment that applied a syllabus proposed by the author, using active methodologies to teach the Exploratory Test Design and Execution being applied and testing the approach in a systematic way. The teaching plan elaborated upon and used as part of the syllabus is based on “learning by doing” in an interactive way, that is, providing student-centered teaching in order to make such an approach more beneficial to the student and, conditioning them to obtain the expected competences in the industry. Methods: The strategies to carry out the research were as follows: (i) a review of the literature to identify gaps in the knowledge about activities performed in the exploratory testing process; then, to elaborate upon the syllabus proposed, (ii) equivalence mapping was carried out on some guides and curricula related to the research theme; (iii) a survey was conducted to identify the tools, techniques, and work products usually used by professionals in the industry; (iv) a teaching plan, based on active methodologies, was produced using the proposed syllabus; (v) then, for the application of this teaching plan in an experiment, a statistical analysis was performed using the two-tailed Student-t approach for independent samples. Results: For this, both groups (experiment and control) were analyzed using the two-tailed Student-t technique, and we noticed that significant statistical gains were reached with the use of the proposed approach, compared to the control group. All participants reported that some basic test concepts, if adopted in their courses, could contribute positively. The participation made them (students) more active in the classes provided by active pedagogical practices, especially when remote teaching was applied; this broke down barriers to communication with their classmates. Conclusion: The results show that the use of active methodologies can provide benefits to the learning of Exploratory Test Design and Execution systems as they allow the application of tools in line with real scenarios observed in software development companies; these results also encourage new studies to explore these ideas further.

Keywords: software testing; test design and execution; exploratory testing; active methodologies; active interventions; pedagogical practice; syllabus; teaching plan; software engineering education



Citation: Costa, I.E.F.; Oliveira, S.R.B.; Elgrably, I.S.; Guerra, A.d.S.; Soares, E.M.; Costa, I.V.F. Using Active Methodologies for Teaching and Learning of Exploratory Test Design and Execution. *Educ. Sci.* **2023**, *13*, 115. <https://doi.org/10.3390/educsci13020115>

Academic Editor: James Albright

Received: 22 November 2022

Revised: 16 January 2023

Accepted: 17 January 2023

Published: 21 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Agile software testing approaches have become protagonists in the industry, with the Exploratory Testing method being considered the ideal approach to be applied in projects that require little time to carry out the activity, as well projects in which managers require rapid and continuous feedback from the testing process [1,2]. However, it is still understood by many professionals as an informal approach, without any structured and organized procedures, and thus many do not support the use of test process management activities [3–6]. Testing in agile teams has been considered not only the functional aspect, but several types of testing related to exploratory testing have been considered suitable to complement other testing approaches; for example, exploratory testing is commonly used to improve test case coverage with a predefined roadmap, and a system under test model is used as a flow map for model-based automated testing [6].

It commonly requires the development of knowledge in the functions inherent to agile teams, mainly in practicing the principles of agile testing, where it values effective communication and cooperation between stakeholders. In addition, testers need to know not only about testing or quality assurance, but also about other subjects, such as business analysis and coding. These are also factors that encourage this study to use an educational approach that promotes greater interaction among students. However, one of the great challenges faced in Software Engineering education is meeting the requirements teaching methods that make this process more effective; this teaching is not restricted to the academic scenario, as large companies also face this problem when developing training for their employees [7].

In the educational context, some forms of teaching that are prescribed in the literature may be alternatives to the classical teaching method, and this work focuses on the use of a set of active methodologies. According to Prince [8], active methodologies consist of pedagogical practices that can stimulate greater interaction, communication, learning involving more practical activities and motivation in students; that is, they are student-centered teaching and learning approaches, where the teacher becomes one more facilitator of this process. Some examples of active methodologies are gamification, project-based learning, problem-based learning, flipped classroom, hybrid learning, etc.

In the State of Testing [1] report, it was emphasized that, in the next three years, a professional with skills in the test planning and application of the exploratory testing approach is expected; in other words, this knowledge has become recurrently requested in the hiring of these quality assurance professionals in the software industry. With this, it is emphasized that alternative pedagogical didactics make active methodologies an interesting strategy to be used to assist a teaching and learning process on the exploratory test approach, given its increasing importance in the software industry.

In this context, the main objective of this work is to present the learning gains from the quantitative analysis of the results of the experiment when adopting active methodologies. The formal experiment was carried out with students of an undergraduate course in Computer Science, where two groups (experimental and control) used the same syllabus. It is emphasized that the research is extensive, where a syllabus for teaching and learning is proposed; there is a focus on carrying out, in a systematic way, the design and execution of activities for exploratory software testing, using active pedagogical practices to provide the student with a role in this learning process. For this, an elaborate teaching plan is an instance of the syllabus, which is applied to statistically observe its effectiveness.

In the context of this research, it is understood that the syllabus is defined as a set of interrelated activities, managed in an organized manner with the aim of achieving the established objectives (competences) [9].

Therefore, the main research question (RQ) of this study is: *Will students be able to acquire more skills related to design and the execution of activities for exploratory software testing, if the learning process involves active pedagogical practices instead of a traditional approach?* This question was established in an attempt to refute the following null hypothesis:

H0 . *The use of the syllabus proposed for teaching and learning exploratory software testing does not provide more basic skills to the student than traditional teaching*

H1 . *The teaching approach using the proposed syllabus allows the student to obtain more skills, compared to the use of the traditional approach*

This research question unfolds into four other questions and their corresponding hypotheses, in order to analyze each teaching unit adopted in the syllabus individually. Thus, it will allow this study to observe, in detail, the effectiveness of learning using the proposed syllabus, compared with the results obtained from a traditional pedagogical approach. It is noteworthy that this study is intended to improve learning in regard to testing activities, as well as provide structured training that can be administered in the academic or industrial scenario.

It is believed that engaging pedagogical practices that can better prepare students for obtaining basic and sufficient skills for the job market should be adopted. Such a teaching strategy should provide skills, based on practical activities aligned with what is used in the industry, that allow the student to be able to carry out test design and execution activities in the software industry [10]. According to Scalaton et al. [11], there is still a considerable deficiency in the teaching of software testing, and instead an intensified focus on theory is reported, which thus lacks practical activities. Therefore, the importance of having a syllabus to help prepare for courses or training on the exploratory test approach is clear, in order to prepare students in practice.

In addition to this introductory section, this article is structured as follows: Section 2 is the literature review, showing the main concepts and related works, Section 3 presents the research methodology, Section 4 presents the teaching of exploratory testing design and execution to software engineers, involving mainly the proposed syllabus, Section 5 presents the teaching plan approach, Section 6 details the teaching plan evaluation strategy, Section 7 details the quantitative analysis of all the data, Section 8 contains the results and discussion of these data, Section 9 describes the threats to the viability of the studies and Section 10 presents the conclusion and some future works.

2. Literature Review

The main concepts involved in this study are described in this section to understand the essential details of the theme.

2.1. Exploratory Test Design and Execution

Exploratory Testing consists of the occurrence of learning, test design and test execution simultaneously, that is, tests are not defined in advance in a pre-established test plan, but are dynamically designed, executed and modified where the effectiveness of exploratory testing depends on the software engineer's knowledge [9,12–14]. Recently, Bach [3] defined exploratory testing as an approach to testing that consists of evaluating a product by learning about it through exploration and experimentation, including, to some degree, questioning, studying, modeling, observing, and inferring, among others.

Bach [3] emphasized that the exploratory test is a formal and structured approach, exemplified by the analogy with the taxi ride case, where the customer does not request the ride plan because he trusts the intentions and competence of the taxi drivers. The same happens when using exploratory testing, where the tester trusts the exploitation strategies implicitly adopted. It is noteworthy that some studies identified that testers implicitly apply several exploration strategies, depending on their education level [15].

In this context, Hendrickson [16] describes the learning process for the exploratory test approach being applied as Kolb's Learning Cycle (as can be seen in Figure 1). In this case, the tester holds experiences, then makes his inferences, and, therefore, abstracts what he deems relevant and what he manages to perceive about the test object, thus actively experimenting the object. Thus, Lorincz et al. [17] presented results that show that

this Experimental Learning was essential to guide or organize the thinking of testers to detect defects.

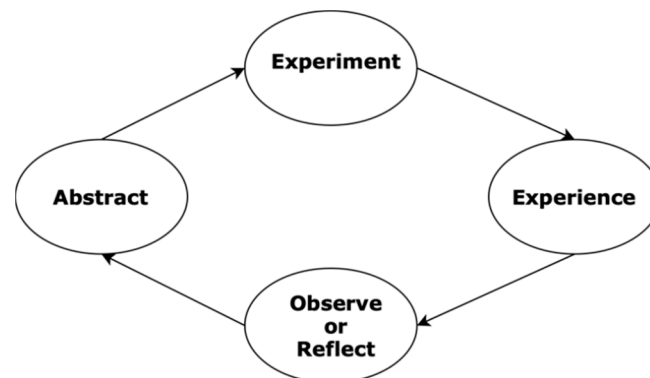


Figure 1. Kolb Learning Cycle [16].

Therefore, it is noticed how the dynamics of the exploratory test process, involving Kolb's Experimental Learning, can be structured (as can be seen in Figure 2). This process starts when the tester elaborates a charter, then it proceeds with the identification of variables and useful heuristics for observing the data necessary to explore the object under test (SUT). For this, the tester conducts an experiment on the SUT, also observing how the object can be explored. Finally, the test manager, technical leader or another tester can produce questions. In this case, they ask about the results, strategies used, and other factors deemed relevant for the knowledge of stakeholders. This feedback moment is important, as it allows them to identify the test coverage explored and the opportunities for improvement.

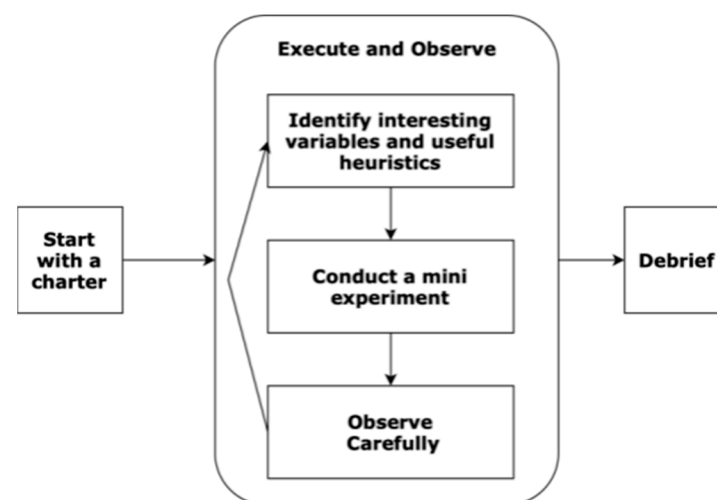


Figure 2. Cycle Structured exploratory testing process involving Kolb's Experimental Learning [15].

In view of this, the possibility of using different exploration strategies (navigation through the test object) to make the tests more efficient is also mentioned [13,16,18]. In addition, Bach [19] reports that, because there are some deficiencies that directly affect the management of test processes, techniques have emerged to mitigate them, for example, Session-Based Test Management (SBTM), Thread-Based Test Management, Risk-Based Test Management. In this case, these management techniques establish more systematic procedures, in order to provide a structured approach to the exploratory testing process, managing to address several factors relevant to the effectiveness of the testing process.

2.2. Active Methodology

New pedagogical practices have often been caused by the insertion of computers and Internet access as an educational resource. Consequently, the need to obtain teaching alternatives that provide better learning, generally using more engaging strategies compared to traditional teaching, has arisen. The importance of pedagogical practices centered on the student has been emerging in debates since the last century, from the movement called “New School”. This idea is based on “learning by doing” in experiences with educational potential, which is notable in active methodologies [20–22].

Active methodology is characterized by the interrelationship between education, society, culture, politics and school, being developed through active and creative methods centered on the learner’s activity in order to promote learning [23,24]. It is a way of conceiving education that presupposes the activity, where the student becomes the protagonist and takes responsibility for his learning process, with the teacher only being a guide in this process [20–22].

When planning to use active methodologies, some questions permeate the understanding of the real role of the teacher. For this, Prensky [25] exposes aspects that better characterize the role of the teacher and the student in an active pedagogy for digital learners (see Table 1). Finally, it is mentioned that there are numerous techniques or methods associated with active methodology. Such methods are as follows: a flipped classroom, project-based learning, problem-based learning, a shared classroom, blended learning, game-based learning (gamification) and so on. It is emphasized that the teacher has the autonomy to adapt or create methods, while respecting the principles of active methodology.

Table 1. Comparison of roles between Teacher and Student in active methodologies.

Teacher	Student
Does not speak, ask	Does not take notes, look for, find
Suggests topics and tools	Searches and finds solutions
Learns technology with students	Learns about quality and rigor with the teacher
Evaluates student solutions and answers, examining quality and accuracy, contextualization	Refines and improves answers, adding rigor, context, quality

2.3. Related Works

In the work of Portela [26], an iterative model for the teaching of Software Engineering (SE) is presented, based on student-focused approaches and practices present in the software industry; this presents significant results that encourage further studies in this context. However, this author focused on the comprehensive teaching of the Software Engineering subject, adapting training practices adopted by the software industry to the academic context, in order for students to develop technical skills in SE at the application level. The author builds the proposed framework based on the analysis of the ACM/IEEE curriculum for computing, the CMMI-DEV quality model and a survey applied to students and teachers.

The work of Furtado and Oliveira [27] presents an approach to the teaching of statistical process control in higher computing courses. It is noteworthy that the author uses the teaching framework proposed by Portela [26] as the basis of his teaching approach, and he also achieves excellent results in the use of active methodologies directed to the specific teaching of statistical process control. The aforementioned author proposes his teaching approach based on the analysis of curricular guidelines for computing, SBC and ACM/IEEE curricula, quality models, as well as a survey applied to teachers, students and software engineers.

The work of Elgrably and Oliveira [28] presents the use of active methodologies aimed at the comprehensive teaching of the subject of software testing, where the applied teaching plan is based on the proposed curriculum from the analysis of national curriculum guidelines, ACM/IEEE and SBC curricula. As with the other studies mentioned above, the

results obtained were quite relevant and encourage the use of pedagogical practices that make the student the center of the teaching–learning process in the context of the Software Testing subject.

Cheiran et al. [29] presented an experience report that compares teaching in the traditional style with the student-centered teaching approach being carried out in the Software Engineering course at the Federal University of Pampa. For this student-centered teaching, the Problem-Based Learning methodology and gamification elements were used; however, in both cases, the approach focused on the teaching of Software Testing topics, and the subject also included requirements engineering, verification and validation topics. It is noteworthy that the referred approach was based on the above-mentioned course syllabus, and the results that were achieved show an effectiveness of teaching using active learning methods; this is because they favored an encouraging environment, according to students' reports and teachers' perceptions.

Lorincz et al. [17] also presented an experience report on teaching software testing, based on gamification. Among the testing approaches, it is emphasized that the main concepts are taught using different active techniques. Thus, the agile testing concepts and session-based test management are learned through Lego-based contexts. For the teaching of exploratory testing through data-based games, and for the teaching of exploratory testing, the Kolb's experimental learning cycle structure was adopted. As a result, there are reports that the use of games motivated the students to participate more intensely in the activities, in which the reflection of their actions allowed their self-discovery about the concepts encapsulated in the games. In view of the results, it is essentially emphasized that most students agreed that Experimental Learning created a better understanding of how to conduct exploratory tests.

Afshar [30] explores effective approaches for creating a hands-on activity-based curriculum for Software Engineering (SE) so that students can develop strong fundamental SE knowledge and practical skills through the use of hybrid-PBL and integrated learning. The results of the study indicate that hybrid PBL is a practical pedagogical approach that enables students to acquire best practices. Meanwhile, the application of continuous projects for software development in integrated learning satisfied the students' expectations, as they were able to experience circumstances close to reality, including the difficulties observed in the work products generated in this project. Therefore, it is observed that the study presents results for software engineering in general and realizes the focus on software development. Thus, this present work is distinguished by proposing a practical approach using agile methodologies, but aiming at the education of activities essential to the quality of the software and the development process.

In Dorodchi et al. [7], a curriculum for the computer science course is proposed, which is intended for students who have completed the introductory course in programming and data structure. The focus of the course is to provide basic knowledge about software engineering as a software development methodology in a real scenario. As in the work of Afshar [30], the focus is on software development. Thus, it is emphasized that this present work is different because it involves an approach with structured procedures that encompass several active methodologies, including a continuous project.

Fraser et al. [31] conducted a study in which they integrated Code Defenders as a one-semester activity in an undergraduate university course in software testing. Weekly Code Defenders sessions were applied, addressing challenges on how to select the suitable Code for testing, manage games and evaluate performance. The results of this integration of Code Defenders satisfied the students, providing greater engagement in carrying out the tests; this approach was evaluated as being positive for influencing the constant evolution of the students throughout the semester. It is noted that the present study differs in that it uses several active methodologies and provides targeted learning on Exploratory Test Design and Execution; meanwhile, Fraser et al. [31] focus more on preparing unit-level tests to be inserted during software development activities.

In view of the studies presented, this work has similarities, mainly in its use in the analysis of national and international training curricula, mentioned in the first three aforementioned works [26–28]. Regarding these three works, it is mentioned that they served as a basis for structuring and defining the teaching program, in line with industry practices; this differs from other works by focusing on the teaching of exploratory testing being applied systematically, according to the structure of the specific objectives and practices prescribed in the TMMi [32,33]. Furthermore, it is mentioned that the work of Lorincz et al. [17] was important to have as a basis in the use of the Experimental Learning method, proposed by Kolb.

3. Research Methodology

To achieve the objectives of this study, the methodological procedures were established, as shown in Figure 3.

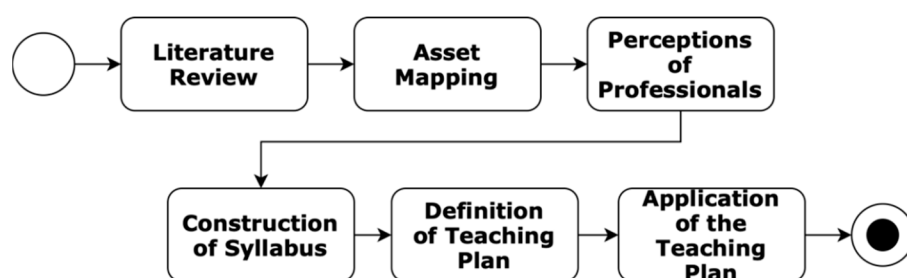


Figure 3. Study Execution Steps.

3.1. Overview Related to the Literature Specialized

A literature review was conducted evaluating the existence of evidence to obtain a holistic view of the strengths and gaps on the applicability of the exploratory testing approach, in the context of the software industry. This research was restricted to studies published between 01/2001 and 04/2020, as this is understood as the first time that the term “Exploratory Testing” has been consolidated [34]. In view of the 20 studies included in the analysis, it was possible to identify a great potential for studying Software Engineering education, with one of the most specific gaps being the non-performance of the structured Exploratory Test Design and Execution activities [35].

3.2. Analysis of Assets from Curricula and Guide for Software Testing

The organizations and their necessary inputs to be analyzed were established. The following were thus identified: (i) TMMi, for providing a guide that describes activities in the Test Design and Execution area, (ii) SBC, for providing curricular guidelines in computing, specifically in the Brazilian context, and (iii) ACM/IEEE, as it also makes available subjects related to computing education at an international level, using both inputs: “Computer Science Curricula 2013” and “Computing Curricula 2020” from ACM/IEEE.

This analysis was based on the specific goals and practices described in the TMMi, being collected as the reference between the assets in the adopted documents, and generating an equivalence structure with the corresponding justifications for each relationship. As a result of this mapping, 13 assets and 110 asset items for Test Design and Execution were identified, interrelated at two levels: (i) Training axes (RF-SBC) and knowledge areas (ACM/IEEE), related to the Test Design and Execution process area, (ii) Derived Content and Competences (RF-SBC), as well as topics and learning outcomes (ACM/IEEE), which relate to the specific goals, specific practices and sub-practices of the TMMi process area, which, in this case, is the focus of this work [36].

3.3. Perceptions of Professionals Regarding Support Resources for Exploratory Testing

The participants were professionals in software testing accredited by a national (Brazilian) and/or international institution. The participants could also be professionals who had

obtained a professional certification in TMMi, with experience in test process improvement. Thus, it is possible to obtain relevant answers to the construction of a teaching plan directed to the Exploratory Test Design and Execution, in order to provide practical content closer to reality [30]. The questions were defined based on the Test Design and Execution process area prescribed in the TMMi to keep the alignment. Further information regarding which questions, the procedures, the method and the planning that were applied in this interview can be observed in the article already published [37].

The results of the interviews were as follows: (i) identification of software tools, techniques and work products for Exploratory Test Design, and (ii) the identification of software tools, techniques and work products for Exploratory Test Execution. In the context of item “i”, it is noteworthy that the risk analysis is one of the most cited activities in relation to support in the identification and prioritization of test conditions; it is also included as being useful as a complement to the Exploratory Test. Regarding work products, the test plan and the results of previous test runs were most used by professionals. In the context of item “ii”, it is noteworthy that the Exploratory Testing with manual and automated strategy was usually used as complementary, while the Incident Report and the Matrix were the work products commonly used by professionals [37].

3.4. Construction of Syllabus

The general competences were defined as being established from the mapping and as based on the competences present in the curricula of the SBC and ACM/IEEE. Subsequently, there was the construction of the syllabus, where there was the organization, structuring and documental record of the teaching and learning approach.

As a result, 12 competences were identified expected from Exploratory Test Design and Execution, making it possible to establish correlated subjects organized into four teaching units that comprise: (i) Exploratory test analysis and design, (ii) Implementation of exploratory testing procedures, (iii) Exploratory test execution, (iv) Test and incident process management. Therefore, some structural elements for each teaching unit of syllabus were defined: prerequisites, guiding questions, programmatic contents, expected results and learning levels, as well as containing the teaching strategy that corresponds to the approach to be used in the teaching plan. All of these were adherent to the learning levels established in the teaching unit [38]. For this, the competences are defined as follows:

1. Employ methodologies that aim to ensure quality criteria throughout the Exploratory Test Design and Execution step for a computational solution,
2. Apply software maintenance, evolution techniques and procedures using the exploratory test approach,
3. Manage the exploratory test approach involving basic management aspects (scope, time, quality, communication, risks, people, integration, stakeholders and business value),
4. Apply techniques for structuring application domain characteristics in the exploratory test approach,
5. Apply techniques and procedures for identifying and prioritizing test conditions (with a focus on exploratory testing), based on requirements and work products generated during software design,
6. Apply software model analysis techniques to enable the traceability of test conditions and test data (with a focus on exploratory testing) to requirements and work products,
7. Apply theories, models and techniques to design, develop, implement and document exploratory testing for software solutions,
8. Apply validation and verification techniques and procedures (static and dynamic) using exploratory testing,
9. Preemptively detect software failures on systems from the exploratory test application,
10. Perform integrative testing and analysis of software components using exploratory test in collaboration with customers,

11. Conduct exploratory testing using appropriate testing tools, focused on the desirable quality attributes specified by the quality assurance team and the customer,
12. Plan and drive the process for designing test cases (charters) for an organization using the exploratory test approach.

3.5. Definition of Teaching Plan

From the syllabus, it was possible to generate this teaching plan containing established learning strategies, a description of the dynamics of activities (methodological procedure) and the expected learning level for each of these pedagogical techniques used (active methodologies). This teaching plan focus involves a student-centered approach, as an alternative to breaking the paradigm of traditional teaching with massive lectures. The identification of possible syllabus contents in the Syllabus Foundation Level, provided by the International Software Testing Qualifications Board (ISTQB), is also noteworthy. This analysis was relevant because it is a study guide commonly used by people seeking the aforementioned certification in software testing [39].

3.6. Application of the Teaching Plan

The experiment was carried out in the first and second semesters of 2021, with the participation of 24 students, initially applied to the control group and, therefore, to the experimental group. Both groups were obtained from 12 volunteer participating students, in order to maintain the most effective comparative analysis by having groups with the same amount of samples, as recommended by [40–42]. Both groups were applied in the subject of software quality, and the control group was conducted by the traditional teaching approach, and the experimental group was conducted from the approach of active methodologies. The study was carried out remotely, due to restrictions arising from the COVID-19 pandemic, where meetings were held twice a week and lasted approximately 2 class hours each meeting. Thus, the content covered in both groups was divided into 4 parts, entitled Evaluation 1, 2, 3 and 4 for the control group, and Teaching Unit I, II, III and IV for the experimental group. Although 24 students actively participated and the results indicated learning gains with the proposed approach, this still cannot be generalized, so new studies must be applied to observe this method's effectiveness.

4. Teaching Exploratory Testing Design and Execution to Software Engineers

As already mentioned in Section 3.4, the 12 basic skills that are needed for a software engineer to work in Exploratory Testing Design and Execution were defined based on the asset mapping. Thus, it is possible to see further details about the syllabus, as follows.

Subject, Syllabus and Goals

Because these skills had been established, it was possible to set out a syllabus that could cover the whole of this background. Thus, the subject-area was divided into four units, as can be seen in Table 2.

The expected level of cognitive ability for each unit and its content was defined by means of terminology, based on Bloom's Revised taxonomy [43,44]. This consists of two dimensions: knowledge and the cognitive process. The possible capabilities of the knowledge dimension and associated verbs are as follows: (i) Factual Knowledge, where the student must be able to master the basic content so that he can perform tasks and solve problems, (ii) Conceptual Knowledge, where the student must be able to understand the interrelationship of the basic elements in a more elaborate context, so the simple elements need to be connected for the formation of knowledge, (iii) Procedural Knowledge, where the student must be able to involve the knowledge of achieving an objective using methods, criteria, algorithms and techniques, thus stimulating the abstract knowledge, and (iv) Metacognitive Knowledge, where the student must be aware of the breadth and depth of the knowledge acquired, so that there is a relationship with the knowledge previously assimilated to solve a given problem.

Table 2. Syllabus Units.

Unit	Goal	Related Basic Skill
Exploratory test analysis and design	Teaching of design concepts and of analysis testing techniques for software design, for the identification and prioritization of test conditions and test data, as well as to present important factors in the definition of an exploratory test process strategy	1, 4, 5, 6, 7, 12
Implementation of exploratory testing procedures management	Teaching how to elaborate charters, possible exploration strategies, preparation procedures for initial tests (intake test) and drawing up a test execution schedule	5, 7, 8, 10, 12
Exploratory test execution	Teaching the exploratory test in practice in a systematic way using structuring techniques, also including the strategies of exploration. In addition, teaching how to write the test reports and apply an analysis of cause	2, 8, 9, 10, 11
Test and incident process	Teaching alternatives to maintain the incident management and important analysis after recording the incidents	2, 3, 12

In relation to the cognitive process dimension, there are the associated verbs: (i) create—produce a new or original work, (ii) evaluate—justify a stand or decision, (iii) analyze—draw connections among ideas, (iv) apply—use information in new situations, (v) understand—explain ideas or concepts, and (vi) remember—recall facts and basic concepts [43]. Appendix A provides a summary of the syllabus, as well as the expected results and the expected learning levels for each topic. The syllabus, in detail, can be consulted in [38].

5. The Teaching Plan Approach

The academic and professional community joins efforts to develop study programs that involve many topics related to computing; these may be sufficient for promoting the skills and abilities of students in order to prepare them for the job market. For this, ACM/IEEE and SBC are essential for presenting structured curriculum guidelines that seek to develop the cognitive potential of graduates, in line with the theory of the area and practical knowledge in the industry. In addition, it is strongly suggested that the adoption of pedagogical practices provides more student engagement than traditional teaching [10,45,46]. In this context, this present study identified and applied some active pedagogical practices to support the teaching and learning of Exploratory Test Design and Execution. These active interventions were as follows: dialogued classes with collaborative learning, problem-based learning, the flipped classroom, team-based learning, and gamification [47–49].

The teaching plan was defined to cover a total of 60 h, divided into 30 synchronous meetings and asynchronously with practical application projects; it was necessary, at times, to monitor the development of the in order to understand and assist the team and streamline the process through mentoring. Faced with the pandemic, and the fact that there are still uncertainties about when to return to face-to-face teaching and what the appropriate procedures to be adopted to return are, it was necessary to resort to a remote teaching strategy. In this way, it was observed that the form of teaching called Education Online (EOL), disseminated by SBC, contains favorable and adherent characteristics for the use of active methodologies through digital technologies.

The EOL educational modality is an alternative to face-to-face classes where didactic-pedagogical practices are supported by the use of many technologies in the networked digital environment. It is a form of distance education, but with characteristics and relationships in the teaching–learning process that are different from conventional Distance

Education (DE) [50]. This EOL is characterized by eight principles according to and defined by Pimentel and Carvalho [50]:

- Knowledge as an “open work”: Knowledge can be re-signified and co-created, being built without limits in a collaborative way;
- Content curation, synthesis and study guides: It describes the act of finding, grouping, organizing or sharing the best and most relevant content on a specific subject;
- Diverse computing environments: Different environments or platforms for interaction between all must be adopted (two-way communication between all);
- Networked, collaborative learning: It is about building knowledge collaboratively, and, in a group, valuing the knowledge of each student in the class with the mediation of a good teacher. In this conception, networked computers are used as a means of social interaction, not as machines for teaching, but rather as connecting people;
- Among all, in interactivity: The exposition of contents by the “dictation of the master”, in which the students can only speak occasionally to clear up a doubt about what was exposed, does not characterize a genuine conversation;
- Authorial activities inspired by cyberculture practices: Practical and authorial activities give the student the opportunity to apply and transform the knowledge of the discipline, giving new meaning to them. Some examples are active methodologies, project-based pedagogy, the flipped classroom, the interactive classroom, and experiential learning;
- Online teaching mediation for collaboration: There is an active mediation aimed at promoting collaboration, in which the teacher is responsible for conducting the dynamics in the group;
- Formative and collaborative evaluation, based on competences: The traces that students leave in digital environments when participating in learning situations designed by the teacher make it possible to carry out an assessment based on competences, valuing not only knowledge, but also skills and attitudes. Online assessment is idealized as a collective action by everyone in the class.

In this case, the teaching plan was conducted based on this EOL, that is, the virtual environment was established following some of these principles related to the context of using educational digital technologies in a remote scenario.

5.1. Exploratory Test Analysis and Design

For this unit, the basic concepts of testing levels, types and techniques are presented to provide a holistic view of Software Testing and allow students to more accurately identify at which point in the software lifecycle the Exploratory Testing approach can be applied. In addition, exploratory test management techniques, test design techniques and work products are exposed for students to analyze and, consequently, support them in the elaboration of charters aligned with the test objective. These techniques are approached with fixation exercises in a practical way, obeying the learning levels and expected results for this topic.

5.2. Implementation of Exploratory Test Procedures

Examples of charters are presented to support the elaboration of new charters for the definition of verification criteria for initial tests in main features of the system, through the collaboration of all. In addition, concepts and examples of exploration techniques are presented for students to correlate with charters. In this case, an activity divided by group is applied in order to encourage more direct collaboration and interaction between the students.

5.3. Exploratory Test Execution

Some practical activities are used to apply the exploratory test in a collaborative way, being dynamized based on Dojo Randori and Dojo Kake (LAB). In addition, the good practices of writing a log or recording an incident, and analyzing the incident in order to

establish the possible cause of the incidents for eligible stakeholders, are presented; the traceability between test conditions, test procedures and test results are always maintained.

5.4. Test and Incident Process Management

In this teaching unit, good practices are presented in relation to the review of incident reports. Subsequently, a paired activity of reviewing the incident log report is applied using the reports generated when searching for incidents (Exploratory Test Execution). In addition, practicing how to write a test summary report follows good practices for effective communication with stakeholders and understanding the possible appropriate incident remediation actions, as well as directing them to closure. Finally, the lessons learned and the opportunities for the improvement of the applied test process are discussed.

It is noteworthy that, for this experiment, the following tools were used as support: Google Meet, e-mail, Instant chat app, Trello, Jira, TestLink, and Microsoft Office.

6. Teaching Plan Evaluation Strategy

To better observe the degree of learning (effectiveness) between the control and experimental groups, four research questions and hypotheses were established, based on the Revised Bloom taxonomy [43,44]. In this context, the results of the scores obtained by the students in the activities related to each teaching unit were used, also considering the interrelationship with the research question submitted for analysis. Thus, the research questions and evaluation instruments involved in the exploratory test teaching units are defined below.

1. For **Teaching Unit I**, the following question and the corresponding hypothesis were defined:

Experiment Research Question 1 (ERQ1): What is the learning effectiveness of the Exploratory Test Analysis and Design unit when adopting the proposed approach, versus the classic approach, at the Evaluate level?

Hypothesis H01: *There will be no difference in effectiveness between the Experimental and Control groups at the Evaluate level.*

For this, the **variables** used were as follows: EF1—Fixation Exercise 1, EF2—Fixation Exercise 2, EF3—Fixation Exercise 3, EF4—Fixation Exercise 4, EF5—Fixation Exercise 5, PP1—Practical Project—Part 1, and PT1—Theoretical Test 1. Thus, the **instruments** used to support were as follows: Fixation Exercises, the Practical Project and the Theoretical Test.

To validate this question, some formulations were necessary; thus, $Ma > Mb$, are the groups, where 'a' is the Experimental Group and 'b' is Control Group. Thus, the experimental group scores are defined as follows:

$$N_{ai} = \frac{((EF1 * 1) + (EF2 * 1) + (EF3 * 1) + (EF4 * 1) + (EF5 * 1) + (PP1 * 2))}{7},$$

where 'i' is a student of Group 'a'.

Thus, the Control Group scores are defined: $N_{bi} = PT1$, where 'i' is a student of Group 'b'.

For this, it was also necessary to calculate the average of the scores of students in Group 'a' and Group 'b', as follows:

$$M_{ai} = \frac{\sum_{i=1}^m N_{ai}}{m},$$

where 'm' is the number of students in Group 'a'.

$$M_{bi} = \frac{\sum_{i=1}^m N_{bi}}{m},$$

where 'm' is the number of students in Group 'b'.

2. In **Teaching Unit II**, the question and the corresponding hypothesis were as follows:

Experiment Research Question 2 (ERQ2): What is the learning effectiveness of the implementation of the Exploratory Test Procedures unit when adopting the proposed approach, versus the classic approach, at the Create level?

Hypothesis H02: *There will be no difference between the effectiveness of the unit in the Experimental and Control groups at the Create level.*

The **variables** identified were as follows: EF6—Fixation Exercise 6, EF7—Fixation Exercise 7, EF8—Fixation Exercise 8, EF9—Fixation Exercise 9, PP2—Practical Project—Part 2, and PT2—Theoretical Test 2. Then, the **instruments** defined were the same as those already listed for teaching unit I. Thus, the **formulations** of the groups were the same as those listed for teaching unit I, with the definitions of the scores also the same as those defined in teaching unit I. Thus, the Experimental Group scores are defined as follows:

$$N_{ai} = \frac{((EF6 * 1) + (EF7 * 1) + (EF8 * 1) + (EF9 * 1) + (PP2 * 2))}{6},$$

where 'i' is a student of Group 'a'.

Then, the Control Group scores are: $N_{bi} = PT2$, where 'i' is a student from Group 'b'.

For this, the average of the scores for students in Group 'a' and Group 'b' is the same as those defined in teaching unit I.

(1) The question and hypothesis defined in **Teaching Unit III** were as follows:

Experiment Research Question 3 (ERQ3): What is the learning effectiveness of the Exploratory Test Execution unit when taking the proposed approach, versus the classic approach, at the Create level?

Hypothesis H03: *There will be no difference between the effectiveness of the unit in the Experimental and Control groups at the Create level.*

The **variables** established were as follows: DR—Fixation Exercise 10 (Dojo Randori), DK—Fixation Exercise 11 (Dojo Kake), EF12—Fixation Exercise 12, PP3—Practical Project—Part 3, and PT3—Theoretical Test 3. Thus, the **instruments** were the same as those defined in teaching unit I and II, including Dojo practice.

The **formulations** of the Groups and the definitions of the scores were also the same as those for teaching unit I and II. Thus, the Experimental Group scores are defined as follows:

$$N_{ai} = \frac{((DR * 1.5) + (DK * 1.5) + (EF12 * 1) + (PP3 * 2))}{6},$$

where 'i' is a student of Group 'a'.

The Control Group scores are then defined as follows: $N_{bi} = PT3$, where 'i' is a student of Group 'b'.

Then, the same average of the scores for students in Group 'a' and Group 'b' were used, the same as those already defined in teaching unit I and II.

(2) Regarding **Teaching Unit IV**, the question and the corresponding hypothesis were as follows:

Experiment Research Question 4 (ERQ4): What is the learning effectiveness of the Test and Incident Process Management unit when adopting the proposed approach, versus the classical approach, at the Evaluate level?

Hypothesis H04: *There will be no difference in effectiveness between the Experimental and Control groups at the Evaluate level.*

For this, the following **variables** were defined: EF13—Fixation Exercise 13, EF14—Fixation Exercise 14, EF15—Fixation Exercise 15, PP4—Practical Project—Part 4, and PT4—Theoretical Test 4. The **instruments** were considered the same as those already listed in teaching unit I and II. The **formulations** of the Groups and the definitions of the scores were the same as those already listed in teaching unit I and II. Thus, the Experimental Group scores were defined as follows:

$$N_{ai} = \frac{((EF13 * 1) + (EF14 * 1) + (EF15 * 1) + (PP4 * 2))}{5},$$

where 'i' is a student of Group 'a'.

Thus, the Control Group scores are as follows: $N_{bi} = PT4$, where 'i' is a student of Group 'b'.

The average of the scores of students in Group 'a' and Group 'b' was defined the same as in teaching unit I and II.

In view of this, the schedule for applying the study to the experimental and control groups is presented. Table 3 presents the number of classes, subjects and pedagogical techniques involved.

Table 3. Experiment schedule.

Day	Control Group	Experimental Group
Inaugural Class	Presentation of the Teaching Plan. Availability of support material.	Presentation of the Teaching Plan. Availability of support material.
Class 2	Applying pre-test	Applying pre-test
Classes 3 to 10	Classical lectures: Teaching Unit I—Exploratory Test Analysis and Design. Test: Evaluation activity with multiple choice and discursive questions about the content taught in the teaching unit.	Dialogued classes: Teaching Unit I—Exploratory Test Analysis and Design. Dialogued classes with collaborative learning, Problem-Based Learning, Team-Based Learning, Flipped Classroom and Gamification.
Classes 11 to 16	Classical lectures: Teaching Unit II—Implementation of Exploratory Test Procedures: Test: Evaluation activity with multiple choice and discursive questions about the content taught in the teaching unit.	Dialogued classes: about Teaching Unit II—Implementation of Exploratory Test Procedures: The same active methodologies listed for Teaching Unit I.
Classes 17 to 22	Classical lectures: Teaching Unit III—Exploratory Test Execution: Test: Evaluation activity with multiple choice and discursive questions about the content taught in the teaching unit.	Dialogued classes: about Teaching Unit III—Exploratory Test Execution: The same active methodologies listed for Teaching Unit I and II, also including Dojo Randori and Dojo Kake.
Classes 23 to 28	Classical lectures: Teaching Unit IV—Test and Incident Process Management: With programmatic contents 4.1 and 4.2. Test: Evaluation activity with multiple choice and discursive questions about the content taught in the teaching unit.	Dialogued classes: about Teaching Unit IV—Test and Incident Process Management: The same active methodologies listed for Teaching Unit I and II.
Class 29	Applying pos-test	Applying pos-test
Class 30	Content perception questionnaire according to Bloom's taxonomy, revised to the unit	Content perception questionnaire according to Bloom's taxonomy, revised to the unit. Feedback Questionnaire.

Each active intervention was applied as described below:

- Dialogued classes with collaborative learning: at the end of unit I, a multiple-choice exercise list was applied with subjects present in the content of the teaching unit, based

on the ISTQB and TMMI certification tests; there were also subjective questions with a moment of discussion about the students' answers and practical activities on the content taught. For other units, subjective questions were asked about the subjects present in the content of the teaching unit, with moments of discussion about the answer and practical activities about the content taught;

- Problem-Based Learning and Team-Based Learning: a practical project was carried out in parallel to the synchronous moments, with the development of the first part of the practical project by teams; at the end of each unit, this was presented in the form of a flipped classroom,
- Flipped Classroom: in the presentation of the practical project at the end of each unit, the students were required to explain the developed product and justify their decisions based on the knowledge they had acquired in synchronous moments or by autonomous investigation in the specialized literature;
- Gamification: some game elements were used to guide the scoring of group tasks and course activities during the experiment,
- Dojo Randori: Challenge of running an exploratory test in a less structured way where some software commonly used by students was used, for example, the institution's own enrollment management software. All the students present participated as a pilot, co-pilot or as the audience. The grade was awarded to all participating students, taking into account the number of complete and correct challenges;
- Dojo Kake: In this practice, multiple teams worked in parallel to carry out the activity in a more systematic way, in this case, performing the Exploratory Test following the guidelines of the SBTM technique and exploration techniques present in the literature. Each team received the same charter and web system as a mission to detect defects. At that moment, the teachers helped, with mentoring on the concepts learned in that teaching unit. Each team received a grade considering the number of defects, and considering the exploitation techniques and strategy they adopted.

7. Data Analysis

This section presents the data obtained from the execution of the experiment, and the corresponding analysis for each research question, RQ1, RQ2, RQ3 and RQ4, defined for this study.

7.1. Analysis of Experiment Research Question 1

In ERQ1, "What is the learning effectiveness of the Exploratory Test Analysis and Design unit when the proposed approach is adopted in relation to the traditional approach at the Evaluate level?", aimed to refute the H01 "There will be no difference in effectiveness between Experimental and Control groups at the Evaluate level", by comparing the two groups (experimental and control) in Teaching Unit I. This analysis considers the normality of the data, in which its objective was to evaluate the difference between both populations within the conditions of varied treatments and two samples (treatment).

The normality of the data was verified by observing the variance between the two populations using the Shapiro–Wilk test. As for its variance, an alternative pattern string (two sided) with a 95% confidence interval was used to calculate the referred variation between the two populations. For this, it was noted that the value obtained for p in the F test was 0.4836, being higher than the significance level of 0.05; this meant there was no significant significance between the observed variances. Therefore, the hypothesis test was performed using Student-t for two equal means and unknown variances, where the p -value is approximately 0.000002622 (2.622×10^{-6}) < 0.05.

This demonstrated a positive effect, representing a possible learning gain in the use of active methodologies in this experimental group, when compared to the control group. Thus, this level of significance observed from the tests provided statistical evidence for H01 to be rejected. Table 4 presents a summary of the data obtained for ERQ1.

Table 4. Comparison of learning effectiveness between the Experimental Group and Control Group for teaching unit I.

Variables	Experimental Group	Control Group
	Evaluation	Evaluation
Sample Size	12	12
Minimum	7	3
Maximum	9.5	6
Sum of Points	100	59
Median	8.5	5
First Quartile	7.75	4
Third Quartile	8.87	6
Average	8.333333	4.916667
Standard Deviation	0.9374369	1.1645002

7.2. Analysis of Experiment Research Question 2

In ERQ2, “What is the learning effectiveness of the Implementation of Exploratory Test Procedures unit when adopting the proposed approach in relation to the traditional approach at the Create level?”, aimed to refute H02, “There will be no difference in effectiveness between Experimental and Control groups at the Create level”, by comparing the two groups (experimental and control) in Teaching Unit II. It is noteworthy that the data were analyzed in the same way as described for ERQ1.

In this context, Shapiro–Wilk test was used to verify the normality of the data observing both populations (Group ‘a’ and Group ‘b’). Thus, it was used two sided with a 95% confidence interval, in order to calculate the referred variation between these populations. In this case, the value for p in the F test was 0.483, being higher than the significance level of 0.05. This meant that there was no significance between the analyzed variances. Therefore, the Student-t for two equal means and unknown variances was used to analyze the hypothesis test, with a p -value of approximately 0.000002654 (2.654×10^{-5}) < 0.05 obtained.

This demonstrates a positive effect, representing a possible learning gain in the use of active methodologies in this experimental group, when compared to the control group. Thus, this level of significance observed from the tests provided statistical evidence for H02 to be rejected. Table 5 presents a summary of the data obtained for ERQ2.

Table 5. Comparison of learning effectiveness between the Experimental Group and Control Group for teaching unit II.

Variables	Experimental Group	Control Group
	Evaluation	Evaluation
Sample Size	12	12
Minimum	7.5	4
Maximum	10	8
Sum of Points	108.5	75
Median	9.4	6
First Quartile	8.375	6
Third Quartile	10	7
Average	9.041667	6.250000
Standard Deviation	0.9159777	1.1381804

7.3. Analysis of Experiment Research Question 3

In ERQ3, “What is the learning effectiveness of the Exploratory Test Execution unit when the proposed approach is adopted in relation to the traditional approach at the Create level?”, aimed to refute the H03, “There will be no difference in effectiveness between the Experimental and Control groups at the Create level”, by comparing the two groups (experimental and control) in Teaching Unit III. It is noteworthy that the data were analyzed in the same way as described for ERQ1.

For this, the normality of the data was guaranteed by analyzing the two populations using the Shapiro–Wilk test. In this context, the variance between the two populations was verified by applying the two sided test with a 95% confidence interval, aiming to calculate that variation. Thus, it was noticed that the value obtained for p in the F test was 0.3563, which, being lower than the significance level of 0.05, represents a significance between the observed variances. It is noteworthy that the hypothesis test was performed using the same as that used in teaching unit I and II, where the p -value was approximately $0.003563 < 0.05$.

This demonstrates a positive effect, representing a possible learning gain in the use of active methodologies in this experimental group, when compared to the control group. Thus, this level of significance observed from the tests provided statistical evidence for H03 to be rejected. Table 6 presents a summary of the data obtained for ERQ3.

Table 6. Comparison of learning effectiveness between the Experimental Group and Control Group for teaching unit III.

Variables	Experimental Group	Control Group
	Evaluation	Evaluation
Sample Size	12	12
Minimum	5	5
Maximum	10	7
Sum of Points	100.5	71
Median	9.25	6
First Quartile	8	5
Third Quartile	9.625	6,250
Average	8.375000	5.916667
Standard Deviation	2.0684316	0.7929615

7.4. Analysis of Experiment Research Question 4

In ERQ4, “What is the learning effectiveness of the Test and Incident Process Management unit when adopting the proposed approach versus the classical approach at the Evaluate level?”, aimed to refute H04, “There will be no difference in effectiveness between the Experimental and Control groups at the Evaluate level”, by comparing the two groups (experimental and control) in Teaching Unit IV. It is noteworthy that the data were analyzed in the same way as described for ERQ1.

It is noteworthy that the Shapiro–Wilk test was important for verifying the normality of the data and variance between the two populations. For variance, an alternative pattern string, named a two sided, was used, considering 95% as the confidence interval to calculate the variation between these populations (Group ‘a’ and Group ‘b’). For this, it was noticed that the value presented for p in the F test was 0.09129. This value was higher than the significance level of 0.05, representing a significance between the variances of these populations. Therefore, the hypothesis test was performed using the same as that used in teaching unit I, II and III (Student-t for two equal means and unknown variances), having a p -value of approximately $0.0004804 < 0.05$.

This demonstrates a positive effect, representing a possible learning gain in the use of active methodologies in this experimental group, when compared to the control group.

Thus, this level of significance observed from the tests provided statistical evidence for H04 to be rejected. Table 7 presents a summary of the data obtained for ERQ4.

Table 7. Comparison of learning effectiveness between the Experimental Group and Control Group for teaching unit IV.

Variables	Experimental Group	Control Group
	Evaluation	Evaluation
Sample Size	12	12
Minimum	7	4
Maximum	10	8
Sum of Points	105.5	74
Median	8.75	6
First Quartile	8	5
Third Quartile	9.625	8
Average	8.791667	6.166667
Standard Deviation	0.9643258	1.6422453

8. Results Discussion

According to the results obtained in this research, it was possible to perceive that the learning effectiveness of the alternative teaching approach presented in this work was superior to that obtained using the teaching methodology with traditional classes, given that hypotheses H01, H02, H03 and H04 were rejected; therefore, the average scores reached by the experimental group during the evaluations reached a greater significance when compared with the control group.

In this context, the use of active methodologies and their adopted practices can be considered one of the factors to achieve these positive results in the experimental group, as the active pedagogical interventions are centered on the student, aiming to promote their engagement and autonomy in the teaching and learning process. It is noticed that there are studies that encourage the use of active methodologies [51–55], but this study differs by using well-founded quantitative methods to evaluate this learning gain with statistical significance. Another factor that maintains the originality of this present study is directly related to the adhering foundation, with international curricula and guides based on industry experiences. This is in addition to the proposal to use several active interventions as a way of streamlining classes and enhancing learning with the application of an entirely practical project, with the active and constant involvement of students.

It is important to highlight that this work focused on comparing only the effects of the use of active methodologies in the Exploratory Test Design and Execution activities; thus, no analysis of the students' performance was carried out. All participants declared that they had not studied or worked with exploratory testing previously, despite having already taken courses in software engineering and been made aware of some basic test concepts, all of which were very superficial. Therefore, it was possible to see that the students had not actually taken any software testing course or taken any course on the exploratory testing approach in particular. Thus, the prior knowledge of the students that could have been related to the subject's content was considered low.

With the qualitative feedback from the students, it was possible to identify that the teaching experience involving active methodologies was considered quite useful and could contribute positively if adopted in their courses. In this context, students reported that their participation made them more active in classes provided by active pedagogical practices, especially in the form of remote teaching; here, barriers are still high when it comes to communicating with others involved in the course, especially when carrying out team activities. This fact corroborates the analysis carried out by Papadakis and Kalogiannakis [56],

where they observed in their research that the use of forms of intervention focused on the student provides a more active participation in classes, unlike the traditional teaching method. Students also reported that this communication and interaction can more actively help them in the real industry environment, by having to interact with unknown people through meetings, asking for help, cooperating with other collaborators, and achieving their goals with a good performance.

Regarding the contents used in the teaching units of the experimental group, students considered them adequate and sufficient for promoting the learning of contents on the basic foundation of software testing; the students obtained the sufficient skills and competences to systematically apply the activities of Exploratory Test Design and Execution not only for the acquired technical knowledge, but also for the abstract skills experienced by the active interaction with the other students (communication, organization, proactivity, empathy etc). The distribution of the content in the teaching units was positively evaluated by the students, as it was possible to dedicate a lot to each stage of the activity carried out, achieving a gradual and evolutionary learning.

These aforementioned factors were no longer mentioned by the control group, possibly because they had a much higher theoretical charge than in practice. It was also observed that, regardless of the form of teaching, students liked more practical activities because they believed that the learning would be better. In other words, this fact generates evidence that “learning by doing” becomes not only an initiative of the teacher or the educational institution, but also of the students themselves, who are able to self-perceive an efficiency in their learning when they combine theory and practice. Another important fact to report was the case of students in the experimental and control groups stating that moments of discussion were important in order to understand the level of knowledge acquired or fixed on certain content until that moment.

It was emphasized that the time focused on theory being greater than the time allocated to practical activities may have had a negative impact on the averages of the students in the control group, as observed in the research questions and hypotheses of this work. Such an impact may have affected the learning of certain key aspects of the subject, where the presence of practice would be more appropriate and generate less difficulty in fixing the content. Practical activities can stimulate engagement, generate student autonomy, and encourage more interactivity in the exchange of knowledge.

Regarding the weaknesses identified in this study, it is worth noting that the COVID-19 pandemic, despite using an approach that involves online digital technologies, still creates many uncertainties. This atypical period made the study even more challenging, particularly when obtaining the maximum attention of students in a virtual classroom, having to wait for proactive, communicative, interactive and cooperative students, and still achieve a sufficient performance to feel apt or skilled enough to apply the knowledge learned in their professional careers. In addition, there are several factors that can arise with remote teaching, for example, internet connection problems, power outages, hardware and software problems, and the lack of the instant observation of students’ attention; this is due to the fact that most prefer not to display their face (local environment) and in some cases, cannot display their face due to the instability of the Internet connection.

Although the experimental group had a greater number of practices than the control group, dialogic expository classes were also used to teach certain content in the teaching plan, especially when there was a need to introduce new concepts or fundamental techniques for use in the activities. However, it was possible to notice that the students dispersed or did not give the expected attention as the more theoretical classes lasted, and that some lost focus and interacted little, corroborating the statements of Branstetter et al. [57] and El-Ali [58], about the duration of the classes. The students reported that the experiment could be improved by reducing some of the expository dialogued classes to a maximum of a one class hour per meeting. In view of this, the application of new studies to determine the most appropriate duration for each expository and dialogued class are intended in the next experiments.

Therefore, it is mentioned that the experiment was applied in 30 meetings lasting approximately 2 h each, totaling 60 h of classes; these were distributed in four teaching units established for the subject of Exploratory Test Design and Execution. It is noteworthy that in the first classes, the students of the experimental group were more withdrawn and interacted less as the practical activities took place, especially in the moments of discussion. The students reported that the dynamics provided by the use of active methodologies allowed them to overcome these initial limitations. The increase in the interactions between students evolved positively, providing evidence that active methodologies were used correctly and that they provided students with more motivation to carry out activities in the classroom [59,60]. In this context, it is noteworthy that the activities of greatest interest to the students were those associated with the teamwork approach; this was one of the items that had game elements, in order to evaluate such interactions and the achievement of the activity's objectives, work organization and deadline delivery. Students even sought cooperation and communication with other groups to help or ask for help about problem-solving alternatives and understand certain content.

Based on these results, it strongly suggests that active learning, when carefully planned and skillfully executed, can help students improve their learning. Such a tendency to obtain benefits can occur in any disciplinary field, in which active learning is not a simple teaching method, but encompasses a range of specific activities; this considers that students are actively engaged in the learning process in the space in which they are immersed.

9. Threats to Validity

This section presents and discusses some threats to the validity of the study. In view of the results obtained in the scientific research, we seek to analyze and interpret them with caution, mainly to establish opinions regarding the generalization of these results. With this, some threats were identified that can influence the results obtained, following the usual classification of threats to validity (internal, external, construction and conclusion).

These threat factors and some actions that can be taken to mitigate them will be presented in the next section. Therefore, it is recommended that the results presented in this study be interpreted within the limits established according to the threats presented below.

9.1. Internal Validity

One of the threat factors concerning the construction of the experimental and control groups involves the participants (students) who enrolled in the Software Quality subject and who were in, at least, their sixth semester of the undergraduate computer science course, for two consecutive semesters. This course is elective (optional) and students participated voluntarily. This was important in order to not influence the formation of the groups and reduce the confounding factor, as well as not cause a possible threat to the statistical regression; this could cause proximity to the control and experimental groups, in order to make them statistically equivalent.

The fact that researchers cannot set limits to students in the search for knowledge from sources external to the materials available makes the internal validity related to maturation a threat. In an attempt to reduce the influence, both groups received the same materials, and the teachers were always available to clarify doubts and help them in asynchronous moments, especially regarding the practical project. For this, the same number of meetings were established with each group, and extra meetings were available at the request of the students.

Another important fact to mitigate the internal threat by instrumentation was the use of an expert to correct the evaluations carried out, not being the teachers who taught the subject to the groups. In addition, the data was granted to a statistical researcher who did not participate in the construction and execution of the subject, with the function of performing the analysis of these obtained data. In this case, the data for the corresponding student was not shared with the experts, in order to mitigate the influence of the evaluation, that is, the students were not known. In this context, a threat involving instrumentation

could also be the fact that the approach used in the subject offered the experimental group different ways of evaluating students between the two groups.

Another factor considered a threat is that the course was taught by two different teachers, one for each group. In this case, the difference in learning can be influenced when observing examples, and the conduction and contribution of the teacher in moments of discussion. Thus, learning gaps between groups can happen, not only because of the approach, but due to the depth of knowledge of the group teacher and their ability to transmit content. In this case, it is noteworthy that the teachers were both a master and a doctor in Computer Science, with more than 5 years of experience researching and working in the industry in this area; therefore, the teachers built the syllabus and the teaching plan together, in order to mitigate this risk.

The proposed syllabus had no similarities with the curriculum currently used in the institution where the research was carried out. Thus, it is relevant to mention that some teachers may not have more specific knowledge on the subject. However, to alleviate this threat, using a set of support materials is recommended for this study, being important and freely accessible to the student and the teacher.

9.2. External Validity

This research used undergraduate students of higher education courses in computing who had not yet taken the subject of software testing, but who had already obtained knowledge about software engineering for a better understanding of Exploratory Test Design and Execution activities. Therefore, it is recommended that the results are generalized through this academic context. For this, the experiment was carried out with a sample of 12 students in each group (experimental and control), where the sample size is considered small and the generalization must be treated with caution due to its limitations. In addition, the study was not applied to other populations, considering several factors; these include the participants' different academic levels and professional experience, and even their desire to work in the area.

This factor of interest in the area can be a factor that can also influence learning, as students reported that they were curious to better understand the approach and the area as a whole, because the course does not have a subject focused on software-testing techniques but require when obtaining software development activities. To mitigate this, testing fundamentals were addressed in the first teaching unit to provide a holistic view of the area and then delve into the systematic applicability of the exploratory testing approach.

9.3. Construction Validity

In view of the results obtained in this research, it is notable that the data collected from samples from the experimental and control groups were small. Therefore, statements and generalizations of the results to larger populations must be interpreted with caution. Therefore, the main validity of the construction comes from the effectiveness of learning between the experimental and control groups, according to the cognition levels of Bloom's revised taxonomy [43,44], where these results may not be sufficient to measure the learning obtained by the participants at each of the established learning levels.

In this context, it cannot be said that students who obtained excellent results, solving questions based on certification content, are able to take the tests and consequently be approved. Another case is that the acquired skills may also not be usable in different scenarios.

9.4. Conclusion Validity

Despite the data collected being enough to indicate a gain in the effectiveness of the proposed approach, the study also considers the number of students, 12 per class, a threat to the validity of establishing definitive conclusions, that is, it is still not possible to generalize the results. Therefore, new experiments have been planned to observe trends and possibly establish a level of effectiveness in relation to the control group. To mitigate this threat, different statistical tests were used for different research questions in this

experiment, considering the variance of data from a given population in the teaching unit under analysis.

Thus, it sought to perform more robust statistical tests, aiming to circumvent the low statistical power of the distribution of data obtained. This may have happened because the experimental group performed mostly group activities, which generated data homogeneity.

10. Conclusions

The objective of this work was to present, in detail, the first experimental results of the teaching and learning approach of Exploratory Software Test Design and Execution, where it was possible to perceive a greater effectiveness of learning in relation to traditional teaching; thus, the main research question was answered. The teaching plan was established from a previously constructed syllabus aimed at the practical application of structured activities of Exploratory Test Design and Execution, being aligned with the practices and goals prescribed in the TMMi. It is noteworthy that this teaching plan involves the use of techniques, software tools and work products commonly used by professionals and prescribed in the specialized literature, as well as the application of active pedagogical practices as an alternative to the traditional teaching and learning paradigm; this includes considering an approach to remote teaching based on a teaching modality (EOL) that also proposes more interactive, communicative and engaging classroom dynamics through the use of digital technologies, unlike the Distance Education modality.

It is emphasized that adherence to the teaching plan is produced from the documents obtained as a basis for analyzing mapping assets, interviews with test professionals, alignment with the TMMi process area and the use of active methodologies through digital technologies. This plan was also preliminarily evaluated by experts, both in research on software engineering education and practical applications in the industry. According to the results obtained in this work, they can be considered significant, as the students participating in the experimental group obtained a good level of learning in the content covered; however, due to threats to validity, the results cannot be generalized in any situation.

Therefore, one of the main limitations existing in this study is related to the sample size, in the case of 12 students in each group (control and experimental), totaling 24 students. However, this limitation did not negatively affect the results, as all students were regularly enrolled in the undergraduate course in Computer Science, having already taken the Software Engineering subject, which was established as a prerequisite, in order to avoid large differences in sampling due to size variation. It is noteworthy that, despite the sample size not being sufficient to generalize the results to the different contexts in which they have been applied, it is a first initiative in order to provide statistically significant results; there will be new positive results in future applications and other studies in the field will be encouraged.

In addition, another important limitation of the study worth mentioning is the involvement of different teachers for each group (Control and Experimental). However, this was also minimized by the fact that these teachers had access to the same set of materials that fully served as a support to prepare and conduct classes within the limits of the established teaching plan. Another factor that contributed to minimizing the differences in learning that could happen with different teachers was the fact that both had the same instructional level and obtained their training in the same research group, carrying out research for more than five years in Software Engineering Education; they had already published renowned papers and articles focusing on software quality, involving academic and industrial scenarios.

As future work, we intend to replicate the experiment in a computer science graduate class to validate its effectiveness with people who are already trained in the area and possibly have professional experiences. Another objective is to use the teaching plan in different academic subjects in computing. In this way, the importance of this study is emphasized in providing training to students at a basic level, sufficient for use in

the industry; this study even breaks the paradigm of understanding that dictates that exploratory testing is an approach only to be free, and also supports new studies in the area.

Author Contributions: Conceptualization, I.E.F.C.; methodology, I.E.F.C. and S.R.B.O.; software, I.E.F.C. and S.R.B.O.; validation, I.E.F.C., S.R.B.O., I.S.E., A.d.S.G., E.M.S. and I.V.F.C.; formal analysis, I.E.F.C., S.R.B.O., I.S.E., A.d.S.G., E.M.S. and I.V.F.C.; investigation, I.E.F.C.; resources, S.R.B.O.; data curation, I.E.F.C.; writing—original draft preparation, I.E.F.C., S.R.B.O., I.S.E., A.d.S.G., E.M.S. and I.V.F.C.; writing—review and editing, I.E.F.C., S.R.B.O., I.S.E., A.d.S.G., E.M.S. and I.V.F.C.; visualization, I.E.F.C. and S.R.B.O.; supervision, S.R.B.O.; project administration, S.R.B.O.; funding acquisition, S.R.B.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical review and approval were waived for this study due to, according the Brazilian Resolution No. 510 of April 7, 2016, the research used the publicly accessible information, and database, whose information is aggregated, without the possibility of individual identification.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data are available upon request.

Acknowledgments: The authors would like to thank CAPES (Coordination and Improvement of Higher Level or Education Personnel) for granting an institutional Doctoral scholarship to the author of this work.

Conflicts of Interest: The authors declare no conflict of interested.

Appendix A

Subject—Exploratory Test Design and Execution	
Prerequisites	
ACM/IEEE: (SE) Software Engineering SBC: Software Engineering	
Guiding Questions	
Q1. How to apply a set of design techniques to identify test conditions and test procedures, both exploratory? Q2. What criteria are suitable for prioritizing test conditions and test procedures, both exploratory? Q3. What aspects are analyzed for the strategic definition of an exploratory testing process? Q4. How to develop test letters (charters) suitable for certain contexts? Q5. How to identify suitable criteria for verification of initial tests? Q6. How to systematically apply exploratory testing considering pre-defined test charts and selected exploration techniques? Q7. How to apply exploratory test session report review best practices? Q8. How to use good practices to communicate incidents to stakeholders?	
Programmatic Content-PC	
1.1 Exploratory Test Analysis and Design	
1.1.1. Test fundamentals 1.1.2. Work products suitable for the analysis and identification of test conditions adhering to the test objective 1.1.3. Test design techniques 1.1.4. Analysis aspects for prioritization of test conditions and procedures 1.1.5. Exploratory test management techniques 1.1.6. Agile testing process (focus on exploratory testing)	
Expected Results	Level of Learning
The student must understand the basics of software quality and testing, including the exploratory testing approach. In addition to understanding the test, design techniques to be used.	Remember/Factual Understand/Conceptual

The student should be able to establish a list of suitable work products for analysis and identification of test conditions and test procedures.	Remember/Factual Understand/Conceptual
The student must be able to identify the test coverage to be achieved with the test design techniques being aligned with established test conditions.	Apply/Conceptual Analyze/Conceptual Evaluate/Procedural
The student must be able to define a suitable test process according to the analysis performed.	Apply/Procedural Analyze/Procedural Evaluate/Conceptual
Programmatic Content-PC	
1.2 Implementation of Exploratory Test Procedures	
1.2.1. Technique and best practices for writing test charts	
1.2.2. Identification of initial test verification criteria	
1.2.3. Exploration techniques	
1.2.4. Test execution schedule development	
Expected Results	Level of Learning
The student must understand and apply best practices for writing test charts.	Understand/Conceptual Apply/Conceptual
The student must be able to analyze and establish verification criteria for initial tests.	Understand/Conceptual Analyze/Conceptual Apply/Conceptual
The student must be able to analyze criteria to enable him to develop an appropriate test execution schedule.	Apply/Procedural Analyze/Procedural Evaluate/Conceptual Create/Conceptual
Programmatic Content-PC	
1.3 Exploratory Test Execution	
1.3.1. Practical execution of exploratory testing using structured exploration and SBMT techniques	
1.3.2. Application of good incident recording practices	
1.3.3. Application of good incident analysis practices	
1.3.4. Application of good practices for maintaining work products as per test results	
Expected Results	Level of Learning
The student must understand and apply exploratory testing, adhering to the pre-defined test charts and selected exploration strategy.	Understand/Conceptual Apply/Procedural
The student must understand and apply exploratory testing, following structured procedures inherent to the SBMT.	Understand/Conceptual Apply/Conceptual & Procedural Analyze/Procedural
The student should be able to understand and apply good practices for recording and analyzing incident causes.	Remember/Factual Understand/Conceptual Apply/Procedural Analyze/Procedural
Programmatic Content-PC	
1.4 Test and Incident Process Management	
1.4.1. Practice reviewing reports exploratory testing sessions	
1.4.2. Good practices for reporting incidents to stakeholders	
1.4.3. Lessons learned analysis for exploratory testing process management	
Expected Results	Level of Learning
The student should be able to understand and apply good practice in reviewing reports and writing a test summary report.	Understand/Conceptual Apply/Conceptual & Procedural
The student must be able to analyze lessons learned to apply appropriate procedures to the management of exploratory testing process.	Remember/Factual Understand/Conceptual Apply/Procedural Analyze/Procedural

References

1. Bhamare, L.; Montvelisky, J. State of Testing: Report 2021. 2021. Available online: <https://www.practitest.com/assets/pdf/state-of-testing-report-2021.pdf> (accessed on 15 January 2023).
2. Parsons, D.; Lal, R.; Lange, M. Test Driven Development: Advancing Knowledge by Conjecture and Confirmation. *Future Internet* **2011**, *3*, 281–297. [CrossRef]
3. Bach, J. Exploratory Testing. SATISFACE, Software Testing for Serious People. 2015. Available online: <https://www.satisfice.com/exploratory-testing> (accessed on 15 January 2023).
4. Elgrably, I.; Oliveira, S. The Importance of Application of Agile Tests in the Software Industry: An Exploratory Approach Using Interview. In Proceedings of the 14th International Conference on Information System and Technology Management-CONTECSI, São Paulo, Brazil, 24–26 May 2017.
5. Pfahl, D.; Mantila, M.; Yin, H.; Much, J. How is Exploratory Testing Used? A state of the Practice Survey. In Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM'14, Torino, Italy, 18–19 September 2014.
6. Gregory, J.; Crispin, L. *More Agile Testing: Learning Journeys for the Whole Team*; Addison-Wesley: Boston, MA, USA, 2015; ISBN 978-0-321-96705-3.
7. Dorodchi, M.; Al-Hossami, E.; Nagahisarchoghaei, M.; Diwadkar, R.S.; Benedict, A. Teaching an Undergraduate Software Engineering Course using Active Learning and Open Source Projects. In Proceedings of the 2019 IEEE Frontiers in Education Conference (FIE), Covington, KY, USA, 16–19 October 2019; pp. 1–5. [CrossRef]
8. Prince, M. Does Active Learning Work? A Review of the Research. *J. Eng. Educ.* **2004**, *93*, 223–232. [CrossRef]
9. Bourque, P.E.; Fairley, R. SWEBOK 3.0: Guide to the Software Engineering Body of Knowledge. IEEE Computer Society. 2014. Available online: <https://cs.fit.edu/~jkgallagher/Schlick/Serious/SWEBOKv3.pdf> (accessed on 15 January 2023).
10. ACM/IEEE. *Computing Curricula: Paradigms for Global Computing Education*; ACM and IEEE Computer Society, Incorporated: New York, NY, USA, 2020.
11. Scatalon, L.P.; Fioravanti, M.L.; Prates, J.M.; Garcia REBarbosa, E.F. A Survey on Graduates' Curriculum-Based Knowledge Gaps in Software Testing. IEEE Frontiers in Education Conference (FIE). 2018. Available online: <http://hdl.handle.net/11449/188892> (accessed on 15 January 2023).
12. Bach, J. Exploratory Testing. In *The Testing Software Engineer*, 2nd ed.; van Veenendaal, E., Ed.; UTN publisher: Den Bosch, The Netherlands, 2004; pp. 253–265.
13. Kaner, C. A Tutorial in Exploratory Testing. QUEST. 2008. Available online: <https://mvt19.lms.nodehill.se/uploads/documents/2019-08-29T09-13-16/Exploratory%20Testing%20-%20Cem%20Kaner.pdf> (accessed on 15 January 2023).
14. Gregory, J.; Crispin, L. *Agile Testing: A Practical Guide for Testers and Agile Teams*, 1st ed.; Addison-Wesley: Boston, MA, USA, 2009; ISBN 13:978-0-321-53446-0.
15. Micallef, M.; Porter, C.; and Borg, A. Do Exploratory Testers Need Formal Training? An Investigation Using HCI Techniques. In Proceedings of the IEEE-Ninth International Conference on Software Testing, Verification, and Validation Workshops, Chicago, IL, USA, 11–15 April 2016.
16. Hendrickon, E. *Explore it! Reduce Risk and Increase Confidence with Exploratory Testing*, 1st ed.; Pragmatic Bookshelf: Raleigh, NC, USA, 2013; ISBN 978-1937785024.
17. Lorincz, B.; Iudean, B.; Vescan, A. Experience Report on Teaching Testing through Gamification. In Proceedings of the 3rd International Workshop on Education through Advanced Software Engineering and Artificial Intelligence (EASEAI'21), Athens, Greece, 23 August 2021.
18. Whittaker, J. *Exploratory Software Testing: Tips, Tricks, Tour, and Techniques to Guide Test Design*, 1st ed.; Pearson Education, Inc.: Hoboken, NJ, USA, 2010.
19. Bach, J. Session-Based Test Management. *Softw. Test. Qual. Eng. Mag.* **2000**, *2*, 32–37.
20. Mattar, J.; Czeszak, W. Avaliação em educação a distância. In *Educação a Distância: Textos Aplicados a Situações Práticas*; Faria, E., Sousa, H., Fernandes, T., Eds.; Gráfica São Matheus: João Pessoa, Brazil, 2013.
21. Madronero, E.; Madronero, M.; Botero, L. Challenges and Possibilities of ICT-Mediated Assessment in Virtual Teaching and Learning Processes. *Future Internet* **2020**, *12*, 232. [CrossRef]
22. García Peñalvo, F.J.; García Holgado, A.; Vázquez Ingelmo, A.; Sánchez Prieto, J.C. Planning, communication and active methodologies: Online assessment of the software engineering subject during the COVID-19 crisis. *RIED. Rev. Iberoam. De Educ. A Distancia* **2021**, *24*, 41–66. [CrossRef]
23. Bacich, L.; Moran, J. *Metodologias Ativas Para Uma Educação Inovadora: Uma Abordagem Teórico-Prática*; Penso: Porto Alegre, Brazil, 2018.
24. Krstikj, A.; Sosa Godina, J.; García Bañuelos, L.; González Peña, O.I.; Quintero Milián, H.N.; Urbina Coronado, P.D.; Vanoye García, A.Y. Analysis of Competency Assessment of Educational Innovation in Upper Secondary School and Higher Education: A Mapping Review. *Sustainability* **2022**, *14*, 8089. [CrossRef]
25. Prensky, M. *Teaching Digital Natives: Partnering for Real Learning*; Corwin Press: Thousand Oaks, CA, USA, 2010.
26. Portela, C. Um Modelo Iterativo Para o Ensino de Engenharia de Software Baseado em Abordagens Focadas no Aluno e Práticas de Capacitação da Indústria. Ph.D. Thesis, Federal University of Pernambuco, Recife, Brazil, 2017.

27. Furtado, J.; Oliveira, S. A Methodology to Teaching Statistical Process Control in Computer Courses. In Proceedings of the 13th International Conference on Evaluation of Novel Approaches to Software Engineering-ENASE; 2018. Available online: <https://www.scitepress.org/papers/2018/68005/68005.pdf> (accessed on 15 January 2023).
28. Elgrably, I.S.; Oliveira, S.R.B. Remote teaching and learning of software testing using active methodologies in the COVID-19 pandemic context. In Proceedings of the Frontiers in Education (FIE) Conference, Lincoln, NE, USA, 13–16 October 2021.
29. Cheiran, J.F.P.; Carvalho, E.L.S.; Rodrigues, E.M.; Silva, J.P.S. Problem-Based Learning to Align Theory and Practice in Software Testing Teaching. In Proceedings of the 31st Brazilian Symposium on Software Engineering, Fortaleza, Brazil, 20–22 September 2017; pp. 328–337. [\[CrossRef\]](#)
30. Afshar, Y. Towards a Practical Software Engineering Curriculum: A Data-Centric Study. Master's Thesis, University of Calgary, Calgary, AB, Canada, 2022. Available online: <http://hdl.handle.net/1880/115286> (accessed on 15 January 2023).
31. Fraser, G.; Gambi, A.; Kreis, M.; Rojas, J.M. Gamifying a Software Testing Course with Code Defenders. In Proceedings of the 50th ACM Technical Symposium on Computer Science Education, Minneapolis, MN, USA, 27 February 2019; pp. 571–577. [\[CrossRef\]](#)
32. Van Veenendaal, E. *Test Maturity Model integration-TMMi Release 1.0: Guidelines for Test Process Improvement*; TMMi Foundation: London, UK, 2018.
33. Van Veenendaal, E. *TMMi in the Agile World-TMMi Release 1.3*; TMMi Foundation: London, UK, 2019.
34. Kaner, C.; Bach, J.; Pettichord, B. *Lessons Learned in Software Testing: A context-Driven Approach*; John Wiley & Sons, Inc.: New York, NY, USA, 2002.
35. Costa, I.; Oliveira, S. An Evidence-Based Study on Automated Exploratory Testing. In Proceedings of the 17th International Conference on Information System and Technology Management-CONTECSI, Virtual, 28–30 October 2020.
36. Costa, I.; Oliveira, S. An Asset Mapping in the ACM / IEEE and SBC Curriculum to the Test Design and Execution of the TMMi. In Proceedings of the Frontiers in Education (FIE) Conference, Lincoln, NE, USA, 13–16 October 2021.
37. Costa, I.; Oliveira, S. A study on assets applied in design and execution activities of exploratory test to be used in teaching-learning: A survey application. In Proceedings of the Frontiers in Education (FIE) Conference, Lincoln, NE, USA, 13–16 October 2021.
38. Costa, I.; Oliveira, S. A syllabus to Support Teaching and Learning of Exploratory Test Design and Execution. In Proceedings of the 14th International Conference on Computer Supported Education-CSEdu, Online, 22–24 April 2022.
39. Costa, I.; Oliveira, S. Development of a Teaching Plan to support Learning Activities of Exploratory Test Design and Execution. In Proceedings of the Frontiers in Education (FIE) Conference, Uppsala, Sweden, 8–11 October 2022.
40. Levanova, E.A.; Galustyan, O.V.; Seryakova, S.B.; Pushkareva, T.V.; Serykh, A.B.; Yezhov, A.V. Students' Project Competency within the Framework of STEM Education. *Int. J. Emerg. Technol. Learn.* **2020**, *15*, 268–276. [\[CrossRef\]](#)
41. Setyarini, T.; Mustaji, M.; Jannah, M. The Effect of Project-Based Learning Assisted PANGTUS on Creative Thinking Ability in Higher Education. *Int. J. Emerg. Technol. Learn.* **2020**, *15*, 245–251. [\[CrossRef\]](#)
42. Tezer, M.; Orekhovskaya, N.A.; Shaleeva, E.F.; Knyazeva, S.A.; Krokhina, J.A. The Effectiveness of STEM Education Applied with a Distance Education Approach. *Int. J. Emerg. Technol. Learn.* **2021**, *16*, 180–192. [\[CrossRef\]](#)
43. Anderson, W.; Krathwohl, D. *A Taxonomy for Learning Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*; Longman: Harlow, UK, 2001.
44. Ferraz, A.; Belhot, R. Bloom's taxonomy: Theoretical review and presentation of the instrument's adjustments to define instructional objectives. *Gest. Prod. São Carlos* **2010**, *17*, 421–431. [\[CrossRef\]](#)
45. ACM/IEEE. *Computer Science Curricula 2013. Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*; ACM: New York, NY, USA, 2013.
46. SBC, Brazilian Computer Society. *Referenciais de Formação para os Cursos de Graduação em Computação*; SBC: State of São Paulo, Brazil, 2017.
47. Akçayır, G.; Akçayır, M. The flipped classroom: A review of its advantages and challenges. *Comput. Educ.* **2018**, *126*, 334–345. [\[CrossRef\]](#)
48. Kalogiannakis, M.; Papadakis, S.; Zourmpakis, A.-I. Gamification in Science Education. A Systematic Review of the Literature. *Educ. Sci.* **2021**, *11*, 22. [\[CrossRef\]](#)
49. Farrow, J.; Kavanagh, S.S.; Samudra, P. Exploring Relationships between Professional Development and Teachers' Enactments of Project-Based Learning. *Educ. Sci.* **2022**, *12*, 282. [\[CrossRef\]](#)
50. Pimentel, M.; Carvalho, F.S.P. *Princípios da Educação Online: Para Sua Aula Não Ficar Massiva Nem Maçante!* SBC Horizontes: Porto Alegre, Brazil. 2020. ISSN 2175-9235. Available online: <http://horizontes.sbc.org.br/index.php/2020/05/23/principios-educacao-online> (accessed on 15 January 2023).
51. Fonseca, J.G.; Brito, C.A.F. Perception of students of the technical course in systems development after experiences with the teaching method peer instruction. *Res. Soc. Dev.* **2021**, *10*, e10710312382. [\[CrossRef\]](#)
52. Andrew, T.; Christopher, L. Using a Flipped Classroom Approach to Support Problem-Based Learning. *Technol. Knowl. Learn.* **2015**, *20*, 299–315. [\[CrossRef\]](#)
53. Chis, A.E.; Moldovan, A.N.; Murphy, L.; Pathak, P.; Muntean, C.H. Investigating Flipped Classroom and Problem-based Learning in a Programming Module for Computing Conversion Course. *Educ. Technol. Soc.* **2018**, *21*, 232–247.
54. de Oliveira, C.M.C.; Canedo, E.D.; Faria, H.; Amaral, L.H.V.; Bonifácio, R. Improving Student's Learning and Cooperation Skills Using Coding Dojos (In the Wild!). In Proceedings of the 2018 IEEE Frontiers in Education Conference (FIE), San Jose, CA, USA, 3–6 October 2018; pp. 1–8. [\[CrossRef\]](#)

55. Wickey, F.; Oliveira, S.; da Costa Carvalho, E. Application of a Teaching Plan for Algorithm Subjects Using Active Methodologies: An Experimental Report. *Int. J. Emerg. Technol. Learn.* **2022**, *17*, 175–207. [[CrossRef](#)]
56. Papadakis, S.; Kalogiannakis, M. Evaluating the effectiveness of a game-based learning approach in modifying students' behavioural outcomes and competence, in an introductory programming course. A case study in Greece. *Int. J. Teach. Case Stud.* **2019**, *10*, 235–250. [[CrossRef](#)]
57. Branstetter, B.F.; Faix, L.E.; Humphrey, A.L.; Schumann, J.B. Preclinical medical student training in radiology: The effect of early exposure. *AJR Am. J. Roentgenol.* **2007**, *188*, W9–W14. [[CrossRef](#)]
58. El-Ali, A.; Kamal, F.; Cabral, C.L.; Squires, J.H. Comparison of traditional and webbased medical student teaching by radiology residents. *J. Am. Coll. Radiol.* **2019**, *16*, 492–495. [[CrossRef](#)]
59. Ribeiro, M.I.; Passos, O. A Study on the Active Methodologies Applied to Teaching and Learning Process in the Computing Area. *IEEE Access* **2020**, *8*, 219083–219097. [[CrossRef](#)]
60. Gonçalves, L.M.; Oliveira, A.M.d.; Ferreira, A.G. The use of active methodologies in teaching for engineering students at a public university, in times of remote activities. *Res. Soc. Dev.* **2020**, *9*, e8819109131. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.