Teaching Exploratory Tests through PBL and JiTT: an experience report in a context of distributed teams

Jarbele C. S. Coutinho
Federal Rural University of the Semi-Arid (UFERSA)
Pau dos Ferros, Rio G. do Norte, Brazil
jarbele.coutinho@ufersa.edu.br

Wilkerson L. Andrade
Federal University of Campina Grande (UFCG)
Campina Grande, Paraíba, Brazil
wilkerson@computacao.ufcg.edu.br

Patrícia D. L. Machado
Federal University of Campina Grande (UFCG)
Campina Grande, Paraíba, Brazil
patricia@computacao.ufcg.edu.br

ABSTRACT

There is no specific professional responsible for quality assurance in agile teams, such as the Test Engineer. Thus, the skills, competencies, and attributions inherent to this professional are the responsibility of all team members. Due to Exploratory Testing (ET) benefits in agile development, there is a need to train agile professionals. In this sense, this paper aims to investigate the contributions and limitations of adopting Problem-Based Learning (PBL) and Just-in-Time Teaching (JiTT) in ET teaching-learning. For this, we conducted a course in remote learning format with agile developers, distributed geographically. Data were collected through an online questionnaire and examined with quantitative and qualitative analysis at the end of the course. Our main findings are that (1) the collaboration between the participants and the adoption of a real problem, along with (2) activities and resources made available before the class, and (3) the existence of specific tool support for ET sessions optimized learning in the context of remote learning.

KEYWORDS

Software testing, Exploratory Testing, Testing Education, Testing Learning and Teaching, Active Learning, JiTT, Just-in-Time Teaching, PBL, Problem Based Learning.

ACM Reference Format:

Jarbele C. S. Coutinho, Wilkerson L. Andrade, and Patrícia D. L. Machado. 2021. Teaching Exploratory Tests through PBL and JiTT: an experience report in a context of distributed teams. In *Brazilian Symposium on Software Engineering (SBES '21), September 27-October 1, 2021, Joinville, Brazil.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3474624.3476977

1 INTRODUCTION

Aligning theory and practice regarding the teaching of Software Engineering (SE) is a persistent challenge, both in the academic context and in the industry [17]. Providing and stimulating experiences that contribute to the technical and non-technical training of students and professionals in this area requires actions to plan the curriculum and curricular components, articulate new teaching methodologies, and include innovative pedagogical elements [6].

SBES '21, September 27-October 1, 2021, Joinville, Brazil

ACM ISBN 978-1-4503-9061-3/21/09…$15.00
https://doi.org/10.1145/3474624.3476977

In this context, the teaching of Software Testing (ST) also stands out. For Cheiran et. al. [6], ST is one of the areas of SE that presents challenges for teaching. It may be difficult and inefficient to teach ST through lectures and lectures. Additionally, the simplicity of the criteria is a factor that makes it possible for ST contents to be part of non-specific subjects, such as SE [24]. Moreover, ST contents may be part of the training provided by companies when their employees do not know a given ST practice or technique.

Among the existing ST practices, we have Exploratory Testing (ET). ET emphasizes the responsibility and freedom of the tester to explore the system, allowing the tester to acquire knowledge of the program in parallel with the execution of the tests [3, 7, 16, 29], as there is no script planning or the definition of test cases defined in test plans [16]. For Bach [3], ET is learning, designing, and executing tests performed simultaneously.

As a way to meet the need for management and measurement of exploratory testing, Bach [3] proposed (1) to divide the testing activities into sessions, which would be the basic unit of work, (2) to stipulate a mission for each session, and (3) adopt time metrics related to testing activities [5], thus giving rise to the Session-Based Test Management (SBTM) approach.

Although the problem associated with ST teaching is being discussed with greater visibility by the academic and scientific community [2, 13, 14, 22, 27] and is producing more specific developments [6, 8, 10, 18, 23–25], few studies investigate the possibilities of streamlining the teaching and application of ET in practice [7, 11].

Adopting more dynamic strategies that bring theory and practice closer together to provide academic-professional training by the real scenario of the software industry is not a trivial task, especially when this experience is conducted with geographically distributed teams that work in an agile environment.

When conducting experiences like this, some challenges emerge, such as (1) integrating the team that works in a cross-functional way, due to the adoption of agile practices; (2) creating conditions for the flow of knowledge to develop, considering the different ways in which people assimilate information; and, (3) dealing with contextual challenges, such as communication, time, internet connection, among others.

There is a need to investigate ways to conduct ET teaching for agile teams working with Distributed Software Development (DSD). Therefore, the question is: *How to encourage practical ET learning with geographically distributed agile teams, seeking integration among members, and promoting active learning?*

In these circumstances, learning in a participatory way, from real problems and situations, can contribute to learning evolution. Problem Based Learning (PBL), is an active learning approach [4, 19] that, from problem-solving, enables students to live experiences that portray the reality of the professional context in the academic environment [6] and aim to encourage the collaborative resolution of challenges through research, reflection and development of solutions.

In an associated way, Just-in-Time Teaching (JiTT) [20] also aims to contribute to student learning. Based on activities carried out before class, JiTT encourages the development of prior knowledge of students [18, 20] so that we can further develop discussions about a given content during class.

This work aims to investigate the contributions and limitations of adopting PBL and JiTT in ET teaching-learning with agile DSD teams in a remote learning context. Thus, it is expected to contribute to the mitigation of the main challenges - mentioned above - faced in the execution of courses conducted in a DSD context, and encourage the adoption of ET in the ST practices developed by agile teams.

In addition to this introductory section, this article is structured as follows: Section 2 discusses an overview of ST Teaching. Section 3 describes the methodological procedure used in this study. Section 4 presents the results obtained, in response to the defined research questions. Section 5 discusses the perspectives, challenges, and limitations of this study, based on the results obtained. Section 6 discusses the threats to the validity of this study. Section 7 exposes the analysis of some related works. Finally, Section 8 presents the final considerations and perspectives for future work.

2 BACKGROUND

In this section, we discuss important aspects related to Teaching Software Testing and Exploratory Testing. We then present and discuss some relevant concepts about two main approaches to active methodologies, PBL and JiTT.

2.1 Teaching Software Testing

ST is an essential activity to guarantee the quality of software. Seeking to meet the need to use teaching methods that make the learning of this activity more effective, some studies have been dedicated to investigating systematic approaches to contribute to the teaching in this area of SE [2, 13, 14, 22, 27].

One of the most significant difficulties for teaching ST is the need to apply the process in practice [8, 22]. At university, sometimes, the teaching of ST is distributed in disciplines in the SE area and does not provide an opportunity for the ST content to be learned in profundity. This aspect causes students to graduate with deficiencies in software testing skills [27].

On the other hand, the industry needs professionals with formation and more solid training in testing. In practice, testing professionals (test analysts, test engineers, or testers) have been looking for options to improve the effectiveness and efficiency of testing [13] both to perform a more effective job and to find better positions in their professional career.

Thus, university graduates and SE professionals self-learn (self-train) ST through books or online resources or by participating in industry training and obtaining certification in the ST area [14], such as those provided by International Software Testing Qualifications Board (ISTQB), for example.

2.2 Exploratory Testing

One type of testing that has become widespread in the agile environment is ET. In this method, test professionals can interact with the system the way they want and explore, without restriction, its functionality [28]. In layman's terms, it can be said that ET allows professionals to learn quickly, adjust their tests, and, in the process, encounter software problems that are often not anticipated in test plans or *scripts*.

For Bach [3] ET is learning, design, and execution of tests performed simultaneously. Thus, the test professional adapts to the system being tested, creates, and improves the tests based on the knowledge acquired during the exploration of the system, without the aid of instructions about the system [5].

In ET, test design and execution are performed at the same time [29]. However, we can perceive some disadvantages in the application of this test. For instance, the lack of preparation, structure, and guidance can lead to many unproductive hours [28]. Also, we can test the same functionality more than once while others are not tested [5], especially when multiple testers or test teams are involved. Moreover, it can be not easy to track the progress of testing professionals [5, 28]; among others.

To overcome some of these disadvantages and as a way of meeting the need for ET management and measurement, Bach [3] proposed (1) to divide the testing activities into sessions, which would be the basic unit of work, (2) to stipulate a mission for each session and (3) adopt time metrics related to testing activities, originating the Session Based Test Management (SBTM) strategy.

The SBTM strategy is used to make ET more effective and with clearer goals [5]. For these reasons, too, ET has gained greater popularity in the agile industry [13, 26, 28], requiring testing professionals to display a little knowledge, experience, and skills with ET. Thus, although Garousi et al. [14] highlight that most courses have trained little about ET, it also recommends more ET coverage in ST education.

2.3 Active Learning: PBL e JiTT

As a way to streamline teaching and offer students differentiated strategies that lead to effective learning, active methodologies emerge as an alternative proposal to traditional teaching-learning approaches [4, 19].

Currently, active methodologies are being adopted in teaching-learning from different areas of knowledge as a way to improve current techniques and involving students in this process [21], not limiting their learning only during class.

Active learning are characterized by stimulating students' autonomy and continuous participation in the learning process [4], through different teaching approaches. Among the modalities of active methodologies that are addressed in the literature are Problem Based Learning and Just-in-Time Teaching.

2.3.1 Problem Based Learning - PBL. PBL is a teaching method that is characterized by the use of problems to initiate and motivate the learning of concepts and promote skills and attitudes necessary for their solution [12]. In addition, PBL also aims to include the acquisition of an integrated and structured knowledge base around real-life problems, as well as promoting group work skills and autonomous learning [6, 10, 12], through collaboration and ethics.

PBL is considered a methodology strongly oriented to processes and accompanied by instruments that can assess its effectiveness [12]. Therefore, the practical immersion promoted by PBL requires a teaching plan. This plan includes well-defined learning objectives, the structuring of a practical environment, the determination of roles for the subjects involved (teacher and student), and result evaluation strategies [6, 12].

2.3.2 Just-in-Time Teaching - JiTT. JiTT is a pedagogical strategy developed by Novak et al. [20], whose essence is to connect activities inside and outside the class through warm-ups [18]. In this approach, students are encouraged to read material about the content of the class and complete a small task online, a few hours before the class takes place [18]. This activity allows the teacher to plan the next class or make considerations in class according to the students' expectations or doubts (answers).

JiTT also aims to encourage students to participate actively in different classroom activities, through greater control over their learning, motivation, and engagement [20]. With JiTT, class time is used more effectively because less time is spent on material that students have learned from reading, and more time is spent on more difficult subjects [18].

3 METHODOLOGY

This research examines the contributions of the use of active methodologies, PBL and JiTT, used in association, to assist in the teaching-learning process of ET, during the application of a course conducted remotely with members of agile teams distributed geographically. Thus, the research is classified as an experience report [30], as it precisely describes: planning, in Section 3.1; the execution, in Section 3.2; and, the analysis procedures, in Section 3.3, as a way to contribute with relevant considerations for the ST teaching area, as well as to allow the replication of this experience in other SE teaching contexts.

To understand the development phases of this experience, Figure 1 illustrates the activities developed, from planning to evaluation.

3.1 Planning

The goals of this experience were defined following the guidelines of the Goal Question Metric (GQM) paradigm. Thus, we seek to **analyze** the PBL and JiTT approach in teaching Exploratory Tests, **with the purpose of** realizing their contributions, **concerning** collaboration and integration between participants of a remotely conducted course,from the researchers' **point of view in the context of** geographically distributed agile teams.

To achieve this goal and conduct this research, we defined the following Research Question (RQ): *"How to encourage practical ET learning with geographically distributed agile teams, seeking integration among members, and promoting active learning?"*.

Thus, RQ aims to identify the main contributions and limitations of the implementation of active learning PBL associated with JiTT in an ET course in remote format, about content learning, integration, collaboration between participants, practical activities, and other aspects inherent in solving problems based on real scenarios. Thus, to answer this RQ a course on ET was planned and executed (see Section 3.1.1 and 3.2) with an agile DSD team. And, in the end,

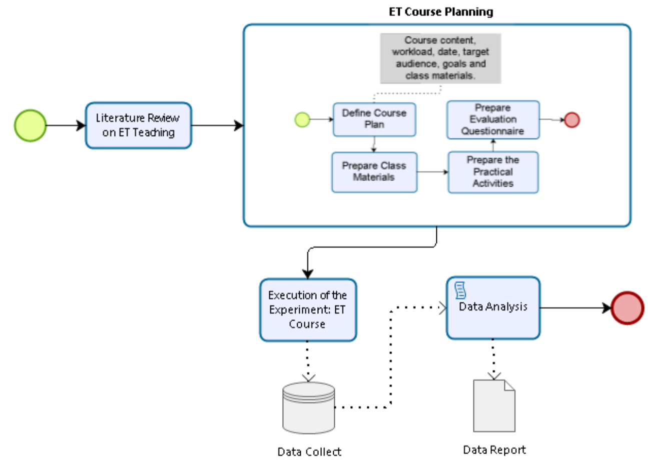

Figure 1: Development phases of this experience.

an online questionnaire was applied to collect the participants' feedback on the adopted teaching-learning methodology.

3.1.1 ET Course Planning. As shown in Figure 1, the planning phase of the ET course consisted of four well-defined steps, described below.

Step 1. Define the Course Plan. In this stage, we defined the course syllabus, the number of hours to be taught, the date of the course, the target audience, the objectives to be achieved, the materials needed, and classes to be produced in a detailed manner according to the adopted methodology. It is important to remark that the definition of this Course Plan was widely discussed, reviewed, and evaluated by two specialists in the ST field. Moreover, we defined the tools to be used in the course, considering the context of remote learning as the following: *Google Meet*, for video communication during classes; *Discord*, for communication between participants during practical activities; *Google Drive*, for storing and sharing class materials and resources (in documents, spreadsheets, and presentations); *Google Forms*, for the elaboration and availability of the evaluation questionnaire, after the course; and, the *Xray Exploratory App*[1], for ET planning and execution, only in the last practical activity.

Step 2. Develop class materials. The classes in this course are intended to train participants on the subject of ET in the agile context and balance the level of knowledge among all agile professionals participating in the course. In this context, the content covered in the class materials was based on [3, 5, 9, 15, 16, 29] and on current lectures, conducted by renowned experts in the field of ET. It is important to highlight that (1) lecture notes (slides) with theoretical content and practical examples on ET were prepared and (2) a handout with a detailed synthesis of the content covered in the course; as well as, we selected (3) a list of tools that support ET planning and execution, (4) a list of videos (tutorials and lectures)

[1] *Xray Exploratory App*: https://www.getxray.app/exploratory-testing-app/

available on the web, and (5) a list of technical articles and books on ET in the agile context.

Step 3. Develop practical activities. To exercise and reinforce learning about the content taught in each module of the course, examples and practical activities were prepared, based on the guidelines provided by the PBL and JiTT methodologies. At this stage, the materials and resources needed to carry out these activities were defined and elaborated, for example: the selection of the web system to be tested; a guide with basic guidelines for each practical activity; templates of the test artifacts (such as Charters, Test Points and Session Report) to optimize the time devoted to each activity; requirements artifacts (such as a System Requirements Specification Document and a Use Case Diagram); and, selecting an installation manual for the *Xray Exploratory App* program. Some of these materials and resources needed to be improved during the course, due to the JiTT methodology applied.

Step 4. Elaborate on the evaluation questionnaire. To collect information about the experience and learning of the participants, a questionnaire was created online[2], with objective and subjective questions. A total of 41 (forty-one) questions were included, distributed between 39 (thirty-nine) multiple-choice questions and 02 (two) open questions, whose answer was optional for the participant.

The questionnaire was designed in *Google Forms* and organized into four sections. So, the first section aimed to briefly characterize the professional profile of respondents. The second section sought to identify the organizational procedures and practices about the ST practice in the Sprints of the projects developed by the agile teams before the course was offered. The third section sought to identify the respondent's perceptions about the teaching-learning obtained during the ET course. Finally, the fourth section aimed to identify the contributions of PBL and JiTT in conducting the ET course.

It is important to highlight that to answer the questionnaire, participants should: (1) have participated in all modules of the course; (2) have carried out the practical activities developed in each module; and, (3) right at the beginning of the questionnaire, have agreed to a Free and Informed Consent Form (FICF) for the research.

In Table 1 the structure of the course is presented, together with the description of the topics and contents covered in the syllabus, and the practical activities planned for the end of each class module. Additionally, the workload defined for each module of the course is informed. Material for this course can be accessed at: https://url.gratis/oKDzKc.

Table 1: Structure of the Exploratory Testing Course

Contents	Practical Activity (PA)
Module I - Introduction 1.1. What are ETs? 1.1.1. ET Characteristics 1.2. What are not ETs? 1.2.1. Randomness and Testing Ad Hoc 1.2.2. Scripted tests 1.3. When to use ET?	PA1 Goal: Understand the product, create hypotheses and plan test scenarios.
Module II - ET in practice 2.1. ET Heuristics 2.2. ET planning 2.3. Writing ET Cases: Charters 2.4. Introduction to SBTM 2.5. Running Tests Based on Sessions 2.6. Evaluation of a session	PA2 Goal: Investigate Heuristics, run tests and log failures. PA3 Goal: Apply Task Breakdown Structure (TBS) metrics.
Module III - A little more about ET 3.1. Problems, Challenges, Solutions 3.2. ET good practices 3.3. ET Support Tools	PA4 Goal: Practice using the XRay Exploratory App tool through the execution of an ET Session.

3.2 Execution

The population of this study included twelve professionals from the software development industry who work with agile methodologies, in the same organization. Currently, these professionals work in geographically distributed locations, due to the *Corona Virus Disease* (Coronavirus Disease) or *Sars-CoV-2* pandemic, which specifically affects the Brazilian population since February 2020. For this reason, too, the ET course was conducted in a completely remote teaching context. In general, the execution of the experience took place as internal training with agile teams of that organization and as planned, in four virtual meetings, with a duration of 02 hours each meeting, on the dates of 06, 07, 12, and 13 April 2021. It is important to highlight that Module II was divided into two meetings, due to the extent of the content taught.

At each meeting, the content was taught and participants were able to ask questions and resolve their doubts throughout the class. Then, to exemplify the discussed theory, a demonstration was made with real examples. And then, participants were instructed to exercise the knowledge obtained through a practical activity based on a real web system. For this, some guidance on the activity was provided.

Participants were distributed in teams and encouraged to interact and collaborate, through the dynamics of each activity. The resolution of a real problem also sought to encourage participants to research, reflect and develop ET relevant to the context analyzed in the activity. This strategy was based on the guidelines provided by the PBL.

At the end of each meeting, class materials and resources were made available to participants so that they had a prior knowledge of the next content to be discussed in the course. This strategy, based on JiTT, sought to encourage interaction between the teacher and the course participants, in addition to enabling more in-depth discussions during the class and anticipating feedback on the materials and resources adopted for the next meeting.

At the end of the course, participants were instructed to fill out a questionnaire online, whose purpose was to collect information about the experience and learning about ET through PBL and JiTT practices.

3.3 Analysis Procedures

After data collection, through the online questionnaire, individual reports were generated, according to the objective of each section

[2] Access to the evaluation questionnaire: https://cutt.ly/Ym5vEk2

investigated in the questionnaire. It is worth noting that the information in these reports was anonymized to preserve the identity of the participants,

Thus, to analyze the data extracted from the content of the responses provided by the participants, a quantitative analysis was conducted [30], mainly in the responses provided through the Likert Scale, with options from 1 to 5 (being: 1 - Totally Disagree; 2 - Partially Disagree; 3 - Neither Agree nor Disagree; 4 - Partially Agree; 5 - Totally Agree). In this sense, the answers were analyzed by class: disagreement, indecision, and agreement. Additionally, a qualitative analysis was conducted on the answers to the subjective questions, but as they were optional or complementary answers to the objective questions, there was little need to apply this type of analysis

4 RESULTS

After the experience was carried out, data were collected and analyzed. In total, the information provided by the twelve course participants was considered, as they all agreed to participate in this study, followed the discussions during classes, and performed all the practical activities provided for at the end of each module.

4.1 Characterization of participants

Initially, to characterize the participants' professional profiles, an analysis was made regarding each team member's attributions and professional experience.

Considering that the composition of agile teams is multidisciplinary, that is, each team member can perform different functions during the developed software project, among the participants in this study, we identified different attributions distributed among the team members (see Table 2), among them are Developer Back-end and Developer Front-end, played by 50% of participants; Software Engineer, 41.7%; Project Manager, 25%; Database Administrator and Tester or Quality Analyst, 16.7% each; Software Architect, Scrum Master, Designer, Mobile Developer and Infrastructure Engineer, with 8.3% each. Other attributions such as Analyst or Business Leader, Analyst or Requirements Engineer, Product Owner (PO), among others, were not informed.

Table 2: Assignments of participants in agile teams.

Assignments	Answers (Nº)	Answers (%)
Database Administrator	2	16.7%
Architect	1	8.30%
Back-end Developer	6	50%
Front-end Developer	6	50%
Designer or Human-Computer Interaction Specialist	1	8.30%
Project or Product Manager	3	25%
Scrum Master	1	8.30%
Software Engineer	5	41.70%
Quality Tester or Analyst	2	16.70%
Mobile Developer	1	8.30%
Infrastructure Engineer	1	8.30%

Regarding the level of academic education of the participants, 58.3% have complete graduation, while 33.3% have a *stricto sensu* post-graduation at the master's level, and only 8.3% are still attending graduation.

Another factor observed was the professional experience of the participants about:

(1) Working experience in the industry software: 50% of them work in this context between 1 and 2 years; 16.7%, between 3 and 5 years; 25%, between 6 and 10 years; and, 8.3%, for more than 11 years. None of them reported little experience with software development, that is, less than 1 year of experience in the market.
(2) Working time with agile methodologies: 50% work in this context between 1 and 2 years; 25%, between 3 and 5 years; 16.7%, between 6 and 10 years; and, 8.3%, for more than 11 years. None of them reported little experience (less than 1 year) or no experience with agile methodologies.
(3) Working time with agile ST: 50% perform tests between 1 and 2 years; 16.7%, between 3 and 5 years; and, 16.7%, between 6 and 10 years. However, another 16.7% reported not working with testing at all.

4.2 Common practices in Agile ST

Additionally, to identify how tests are commonly conducted by agile teams, an analysis of the main ST organizational practices performed in the Sprints of the projects was carried out.

In the context of the participants' professional performance, those responsible for testing the software or the software module developed are: the Back-end Developer (41.70%), the Front-end Developer (25 %), the Product Owner (PO) (16.70%), the Project or Product Manager (41.70%), Scrum Master (8.30%), the Engineer Software (16.70%), the Tester or Quality Analyst (25%) and, in some cases, everyone on the team (41.70%).

We also identified that tests are usually performed throughout the software lifecycle (41.7%). In some phases with more emphasis such as, during (33.3%) or after coding the software (58.30%); and, during (25%) or after the software integration phase (25%). In other phases the test takes place with less intensity such as, during (8.3%) or after the software verification phase (16.70%); during (8.3%) or after the production of software documentation (8.30%); or, during (25%) or after the software maintenance phase (16.70%).

In Agile ST the test types are categorized in Quadrants [9]. Considering this categorization, we notice that the tests performed most frequently by the participants are Unit Tests (50%), Exploratory Tests (50%), Component/Integration Tests (41.7%), Functional Tests (41.7%), Usability Tests (41.7%), Performance and Load Tests (41.7%), Simulations (33.3%), Scenarios (16.7%), User Acceptance Tests (16.7%), Alpha/Beta (8.3%) and Examples (8.3%).

To assess the participants' perception of the types of tests performed on their teams, with regard to your professional activities, we expose the following assertions:

- *Assertive 09:* "I believe that the software testing strategies adopted so far, and reported above, have been sufficient to detect bugs in the system".
- *Assertive 10:* "I believe we need to extend and improve the software testing practices used so far to try to ensure higher quality in whatever product we develop".

These statements contained multiple-choice items, according to the Likert scale, which is detailed in Section 3.3. Table 3 presents the result of the answers to Assertions 09 and 10, and highlights the choices of the Likert scale as follows: (1) Totally Disagree, (2) Partially Disagree, (3) Neither Agree nor disagree, (4) Partially Agree and (5) Completely Agree.

Table 3: Results of responses to Assertions 9 and 10.

	(1)	(2)	(3)	(4)	(5)
Assertive 09	8.3%	25%	8.3%	50%	8.3%
Assertive 10	0%	0%	0%	8.3%	91.7%

Regarding the assertion 9, it was possible to observe a predominance of responses following agreement, which may symbolize that the participants consider the ST strategies used to detect system bugs to be sufficient. Although, in the assertion 10, they unanimously agree that the ST practices, until then adopted by the teams, need to be improved and expanded. These results indicate that although the participants consider the agile testing practices adopted by the team to be sufficient, they also perceive the need to add other ST practices to try to ensure greater quality in the developed projects.

To understand the main problems related to the execution of tests in the projects developed by the participants in the daily work of their teams, we also expose assertions as response options, with multiple-choice items, according to the Likert scale. These assertions can be consulted in the Evaluation Questionnaire (see Section 3.1) and the results of the answers obtained can be seen in Table 4.

The assertions that were most in the agreement were those referring to the constant change in objectives, business process and/or requirements during the Sprint (Assertive c); the existence of hidden, incomplete or inconsistent requirements (Assertion g); lack of knowledge about software testing practices and techniques (Assertive i); lack of training in specific software testing practices and techniques (Assertive j); insufficient time to perform the tests as it should (Assertive k); inexistence of a specific professional, on the team, to perform the tests (Assertive l); and, the effort to perform the tests (Assertive o). Most of the problems highlighted are typical results of teams that work in an agile context [1], and can explain, for example, the need pointed out by the participants to expand and improve the adopted ST practices.

4.3 Perception of ET after the course

We also investigate the learning gained by participants during the course by analyzing the information collected on some key topics in the ET content covered.

For this, we exposed some assertions (see Evaluation Questionnaire in Section 3.1) to the participants and asked that the answers be assigned according to the multiple-choice options, following the Likert scale.

The results associated with assertions 13 to 21 demonstrate a predominant agreement on the learning of all content and practices taught during the course. Among the assertions presented in this evaluation criterion, the following stood out with more emphasis: the importance of simple planning for the execution of the ET

Table 4: Results of responses on problems in ST practice.

	(1)	(2)	(3)	(4)	(5)
Assertive a	50%	25%	16.7%	0%	8.3%
Assertive b	58.3%	16.7%	8.3%	8.3%	8.3%
Assertive c	16.7%	8.3%	8.3%	50%	16.7%
Assertive d	58.3%	0%	16.7%	16.7%	8.3%
Assertive e	58.3%	16.7%	0%	16.7%	8.3%
Assertive f	25%	16.7%	16.7%	25%	16.7%
Assertive g	33.3%	8.3%	0%	33.3%	25%
Assertive h	33.3%	25%	16.7%	16.7%	8.3%
Assertive i	0%	16.7%	25%	25%	33.3%
Assertive j	0%	8.3%	25%	33.3%	33.3%
Assertive k	0%	25%	8.3%	41.7%	25%
Assertive l	0%	0%	8.3%	25%	66.7%
Assertive m	25%	25%	16.7%	16.7%	16.7%
Assertive n	25%	16.7%	25%	8.3%	25%
Assertive o	25%	8.3%	16.7%	16.7%	33.3%
Assertive p	50%	8.3%	8.3%	25%	8.3%
Assertive q	33.3%	25%	16.7%	16.7%	8.3%
Assertive r	25%	8.3%	33.3%	16.7%	16.7%
Assertive s	25%	16.7%	33.3%	16.7%	8.3%

(Assertive 16); that requirements artifacts, even less detailed, can contribute to Session Setup (Assertion 18); the importance of the Alignment Meeting as a strategy to register possible failures, create possible formal test cases, create new missions, register possible requirements, and register new test points (Assertive 21); the usefulness and importance of simple ET artifacts generated in conducting the SBTM are useful for the execution of (Assertive 19); the relevance of defining Heuristics in ET (Assertive 13); among other relevant assertions presented in the Table 5.

Table 5: Results of answers from Assertions 13 to 21.

	(1)	(2)	(3)	(4)	(5)
Assertive 13	0%	0%	8.3%	33.3%	58.3%
Assertive 14	0%	0%	25%	25%	50%
Assertive 15	0%	0%	16.7%	33.3%	50%
Assertive 16	0%	0%	0%	25%	75%
Assertive 17	0%	8.3%	8.3%	50%	33.3%
Assertive 18	0%	8.3%	8.3%	16.7%	66.7%
Assertive 19	0%	0%	25%	16.7%	58.3%
Assertive 20	8.3%	0%	25%	50%	16.7%
Assertive 21	0%	8.3%	0%	25%	66.7%

The performance of practical activities provided participants with a real experience with challenges common to ETs, such as little domain knowledge and necessary qualities of testers in the application of ET (91.7%) makes it difficult to carry out the tests; the absence of an ET plan results in the same functionality being tested several times or an important functionality may not be tested or a serious error may go undetected (91.7%); the lack of a definition of test cases makes it difficult to reproduce the tests performed, if necessary, such as in regression tests (75%); an incorrectly interpreted output can lead to defects that may remain in the system or be eventually detected in future tests (5%); as there is no detailed test guide or plan, and no more complete artifacts are produced other than the crash report, it is difficult to know what has been and has not been tested (50%); and, ET is not suitable for real-time systems (8.3%).

4.4 Contributions from the PBL and JiTT approach

To identify the contributions of PBL and JiTT in the teaching-learning process applied in the ET course, an analysis of the characteristics of these methodologies was carried out. In this perspective, a set of eighteen assertions (23 to 40) were exposed to the participants to be analyzed and answered through multiple-choice options, also following the Likert scale. These assertions can be consulted in the Evaluation Questionnaire (see Section 3.1).

To more accurately classify the answers given, we grouped the assertions correlated to the main practices of active methodologies, in general - perceived in Assertions 23 to 25; PBL, in Assertions 26 to 29, and 36; and, JiTT, in Assertions 32 to 36. We highlight that Assertions 36 to 40, characterized both practices common to PBL and JiTT. Assertions 30 and 31 sought to understand the participants' perception of the dynamics of the course in the remote setting. Table 6 displays the answers given to the assertions.

Table 6: Result of responses from Assertions 23 to 40.

	(1)	(2)	(3)	(4)	(5)
Assertive 23	0%	0%	8.3%	16.7%	75%
Assertive 24	25%	25%	33.3%	8.3%	8.3%
Assertive 25	0%	8.3%	0%	25%	66.7%
Assertive 26	8.3%	16.7%	0%	33.3%	41.7%
Assertive 27	0%	0%	0%	25%	75%
Assertive 28	8.3%	16.7%	16.7%	25%	33.3%
Assertive 29	0%	0%	16.70%	16.7%	66.7%
Assertive 30	0%	0%	8.3%	25%	66.7%
Assertive 31	0%	0%	0%	8.3%	91.2%
Assertive 32	0%	0%	0%	16.7%	83.3%
Assertive 33	0%	0%	8.3%	25%	66.7%
Assertive 34	0%	0%	33.3%	16.7%	50%
Assertive 35	0%	8.3%	8.3%	8.3%	75%
Assertive 36	0%	0%	8.3%	16.7%	75%
Assertive 37	0%	16.7%	8.3%	33.3%	41.7%
Assertive 38	0%	8.3%	16.7%	25%	50%
Assertive 39	0%	0%	8.3%	33.3%	41.7%
Assertive 40	25%	33.3%	0%	25%	16.7%

About general practices guided by active methodologies, we investigated the participants' perception of the inclusion of real practical examples in the activities carried out in the course. It was possible to observe a predominance in the follow-up of agreement in the answers to assertions 23 and 25, which refer respectively to "the scenario (web system) worked on in practical activities represented a real scenario of software development" and "practicing the content theory with a real web system helped to better understand the concepts of ET". However, in assertion 24 we identified a majority of disagreement to the "high level of complexity of the scenario worked on in practical activities".

The answers provided in assertions 26 to 29 and 36, related to PBL practices, which deal with the inclusion of real problems as practical activities in the teaching of content, provide evidence of the efficiency of using this methodology, especially about: learning autonomously and independently about ET (Assertive 26); collaborative group work to expand team discussions on the theory learned (Assertive 27), deliver project activities on time (Assertive 28) and with quality (Assertive 29); encouragement of group work skills, such as distributing roles for each member, setting goals, understanding objectives, providing collaboration and communication, among other aspects (Assertive 36).

About statements 32 to 36, which deal with information about practices common to JiTT, the results provided by the participants express a majority of agreement on the contributions and feedback given to the class from the access of the contents and materials of the class. Thus, from the point of view of the participants, this strategy: contributed to the organization of the class and the instructor's practice in the next class (Assertive 32); helped the instructor to focus on the main difficulties that were expressed by the participants (Assertive 33); and, maximized efficiency and class time (Assertive 34). Another JiTT characteristic observed in the assertions and which showed a predominance of the agreement was related to the objective of practical activities, namely: the encouragement of oral and written communication, through discussions with the team and preparation of test artifacts (Assertive 35) and the encouragement of group work skills (Assertive 36).

Otherwise, we investigated how collaboration between the team in practical activities stimulated the participants' learning. Agreement in Assertions 37, 38, and 39 prevailed, which referred, respectively, to security in collaborating more in the final practical activities than in the initial ones, as it is already better adapted to the business scenario provided as a real example in the activity; security in performing the activities after more specific instructions from the instructor; and, motivation after the instructor provides templates for the test artifacts. A positive aspect was the predominant disagreement in statement 40. A large part of the participants disagreed that they had "problems in collaborating in practical activities because they could not understand them". This result can be explained by the aspects already confirmed in statements 37 to 39.

Finally, the benefits and difficulties of participating in the theoretical and practical activities of the course were investigated, given its implementation in a completely remote teaching context. Table 7 presents the main testimonies of the participants regarding the perceived benefits and difficulties.

According to the statements reported in the responses, we identified that the content, the main approaches, and the ET tools were not known by some of the participants, as well as the usefulness of this test in agile methodologies. These aspects were pointed out as a benefit of the course for the work developed by the teams. Regarding the reported difficulties, we found that the practices could have been conducted with products developed by the teams themselves as a way to facilitate the understanding of the business scenario, and the course load was also considered short for the extension of the content and developed practices.

5 DISCUSSION

To promote active learning and integration among geographically distributed participants during an ET course in a remote learning format, PBL and JiTT approaches have proven to be useful in

Table 7: Benefits and Difficulties of participating in an ET course in remote format.

Benefits
Participant B: "The ET scope and planning to deliver quality software."
Participant H: "Learn how to document the execution of exploratory tests."
Participant I: "Learning about the topic. Although the team, which also works together on development, somewhat adopted what was proposed in the classes, it was clear how we could improve."
Participant J: "The theoretical content and practical activities were of great importance for a more solid understanding of the ET... The requirements document should also be detailed enough to enable the planning of the ET by the responsible team ... In addition, the ET technique presented in the training can contribute a lot to the quality of the developed artifacts."
Participant K: "Through the course, I had my first contact with ET, and participating in it made me learn a lot... What was seen seemed very attractive for the context of agile methodology."
Participant N: "I found the use of heuristics in the tests interesting, I didn't know about it."
Difficulties
Participant B: "Not having any type of ST course in my graduation."
Participant F: "The guidelines of the support material, sometimes it was not clear what should be done, generating doubt in the group."
Participant H: "Differentiate what information should be placed in each field of the template provided during ET planning."
Participant L: "It was a little difficult to think of scenarios for an initially unknown system. I think it would be more beneficial if the practice of the course had used a system developed and well known by the team/class."
Participant O: "I had difficulty participating due to the course schedule, as it wasn't my actual work schedule."

stimulating hands-on learning in this context. According to the agreement of information and reports from the participants, some characteristics of PBL and JiTT stood out, such as:

(1) *The use of a real scenario* of software development contributed to the practice of the ET concepts covered in the course. The actual scenario practiced encouraged the participants to further investigate the possible failures of the analyzed system from the Heuristics and planned ET scenarios. This strategy allowed the quick identification of some bugs implemented in the system's functionalities - in total, there were 6 bugs in three different missions (test scenarios) tested in practice 2, and 4 bugs in three distinct missions tested in practice 3. We highlight that each mission was executed in a 30-minute session. The low level of complexity of the adopted web system also contributed to the understanding of its operation, since there were - in the first practical activities - more detailed requirements or business artifacts.

(2) *Autonomous learning* was stimulated through practical activities, simulating the participants' daily situations through the exploration of the web system; studying or reading classroom material and resources in advance; the discussion of content and activities during classes; the elaboration of questions about the understanding of practical activities; and, the construction of the generated ET artifacts.

(3) *Collaborative work* stimulated different learning styles among the participants, such as: distributing the roles of each member, setting goals, understanding objectives, and providing communication. It also contributed to the expansion of discussions in the team, with the different points of view of the participants, and with the delivery of the activity on time - although, in some cases, additional time to complete the activity was necessary - and with quality - meeting the requirements of the activity. The use of online conversation and collaboration tools contributed to the team's interaction, narrowing the physical distance.

(4) *Stimulation of additional skills*, such as reading materials, using logical reasoning to understand the features of the web system during practical activities, discussions between teams, exploring the system, among others. Teamwork was also an encouraging practice, although the participants already act in this way in their daily work.

(5) *The motivation.* Some clarifications about technical terms, expressions, or ET artifacts were useful to keep the participants motivated in carrying out the practical activities. As well as the availability of ET artifact templates and the socialization of the generated artifacts at the end of each practical activity.

(6) *Feedback* provided before, during, and after classes about the content, materials, resources, and methodology used, contributed to the organization and practice of the instructor in the following class; help the instructor focus on the main difficulties that were expressed by the participants; and, maximize the effectiveness and time of the class.

(7) *The examples shown*, as well as the way to present them, contributed to improve the understanding of TE, as all the examples also referred to contexts of real systems. Exemplifying the theory in this way helped the participants in the understanding and applicability of ET, especially during the performance of practical activities.

It is important to highlight that the Likert Scale helped to identify both the benefits and limitations of the PBL and JiTT approaches, through assertions that represent their main characteristics. However, the answers provided by the participants pointed to these (assertive) characteristics more as benefits than as limitations to the use of these approaches in remote learning. Although guidance and some clarifications were provided during the practical activities, some participants agreed on the difficulty in collaborating in practical activities because they were unable to understand them well. Perhaps this is justified by the absence of face-to-face contact to facilitate communication.

In summary, it is also important to highlight that: *(i)* although during the course, the participants were geographically dispersed, we did not address any specific DSD process. The purpose of the course was to use strategies and tools that would make ET viable in the context of isolated and remote work; *(ii)* the use of a specific tool (Xray Exploratory App) for planning, executing and reporting bugs (with video recording, capturing and annotating screenshots, annotations, among other aspects) in ET sessions are benefits not found in other tools that aid the execution of ET; *(iii)* the experience of remote learning with geographically distributed participants is challenging and factors such as stimulating participation, collaboration and attention skills need to be considered for learning to actually happen; *(iv)* we noticed: the interest and engagement of the participants, when practicing the theoretical content through a real problem adopted in the activities and discussions during the classes, mainly due to the pre-constructed knowledge through prior access to the materials of the classroom; the quality of the answers in the exercises, as a good part of the test artifacts generated were in accordance with the criteria suggested in the description of the activities; a motivation to use the Xray Exploratory App tool, due to the ease in creating, executing and exploring the ET performed; among other aspects already discussed in this Section.

6 LIMITATIONS AND THREATS TO VALIDITY

Some potential threats to the validity of this study were perceived, such as threats to internal, external, construct and conclusion validity. For this reason, some measures were taken to minimize them. To mitigate *construct validity*, the course material and evaluation questionnaire were iteratively planned, updated and validated by the authors. As well as, elaborated based on works related to the ET area, in the context of Agile ST [3, 5, 16, 28, 29]. To mitigate *internal validity* and ensure anonymity of responses, participant identification was optional - via email address. This allowed the data analysis to be performed in an impersonal way. Other aspects inherent to the selection of individuals and conduct of the experiment also contributed and are detailed in Sections 3.2 and 3.3. The *external validity* was attenuated with the availability of resources and teaching materials to facilitate the application of the active methodologies mentioned. Thus, the results are independent of the sample and can be valid for other course participants either in a remote or face-to-face teaching format. To mitigate the *conclusion validity*, only percentages were used to identify common patterns. Complementarily, the questionnaire validation answers were also discarded, regarding possible errors, such as answer format, textual expressions used in the questions, among others. We tried to reduce bias using Likert scale data. Thus, all the conclusions we draw in this study are strictly traceable to the data.

7 RELATED WORKS

Although the literature shows interest in TS Teaching and is seeking strategies to boost practical teaching closer to the real context of the software industry through approaches based on active learning [6, 10, 12, 18], there are still few studies that present results on the teaching of ET [7, 11].

Cheiran et al. [6] present an account of two experiences on the teaching of ST using PBL in an undergraduate course of SE at the Federal University of Pampa (UniPampa). In total, 51 students participated - 25 in the 1st edition and 26 in the 2nd edition of the course. Data collection took place through questionnaires. To analyze the collected data, statistical and content analyses were adopted. The results point to evidence of students' maturity in the context of the curricular component and the benefits and problems faced by integrating PBL and gamification elements.

Andrade, Neves and Delamaro [10] also conducts a study on ST learning using PBL practices, with students from Computer Science, at the University of São Paulo (USP), and Information Systems, of the Federal University of Juiz de Fora (UFJF). The results show that *(i)* classes with many students should have fewer presentations; *(ii)* courses with an average number of students can choose to keep weekly presentations more dynamic or with fewer presentations; *(iii)* the approach PBL is not as effective for students who have less time for extra classes. In summary, it was noticed that the successful adoption of an active approach is not directly linked to the infrastructural aspects.

Figueredo et al. [12] apply PBL to train test engineers. For this, an empirical study was carried out with two groups, where each group was composed of five undergraduate students. The group should test a CASE tool to support functional testing using ET. Two evaluations were made with the participants - one before and one after the execution. Participants' knowledge, grades, and amount of bugs identified were evaluated. The results obtained highlight that the PBL provides the engagement of participants and obtaining experience in scenarios that simulate real ST situations.

Martinez [18] describes the results of an experience with JiTT based teaching in a graduate course in ST over two semesters. The approach adopted was evaluated from the perspective of students, through a survey, and of teachers, from an assessment of strengths and limitations. The results show that a large majority of students (1) believe that their learning has improved when they prepare for class by reading the material in advance and (2) consider JiTT to be an adequate teaching strategy for the course. Teachers highlighted that students became more involved and participatory in discussions during class.

Costa et al. [7] uses gamification as a motivating strategy in teaching and learning ET. This dynamic consisted of practical activity to apply ET in the form of a game, which refers to the "Treasure Hunt". An experience was carried out with students from an SE discipline of an undergraduate course in Computer Science. The results indicate that the qualitative results converged with the quantitative results obtained, showing that gamification helped in the teaching and learning process of the students (forehead pains).

In another work, Costa and Oliveira [11] replicate a new experience with the gamification strategy for teaching ET discussed in [7], with a group of undergraduate students in Computer Science and with graduate students in a Computer Technician course. As a result, students achieved good overall performance. Some reports highlight that gamification facilitated and significantly contributed to better performance, converging with the quantitative data obtained. This can be evidenced mainly by the fact that both "runs" of the experience (classes) reached a percentage higher than 70% of achievement.

In this way, this article differs from the others in that it identifies and discusses the contributions of the integration of active methodologies, PBL and JiTT, in teaching ET in a remote learning course with agile professionals from the software industry and distributed geographically. Some strategies and guidelines seeking to optimize teaching-learning with PBL and JiTT, as well as the discussion of some perceived challenges, were also highlighted.

We could not find works that apply JITT and PBL in ET. In general, the application of JITT or PBL in ST, as reported in the literature [6, 12, 18], achieved results that converge with ours in the sense that the adoption of these methodologies has provided positive gains, related to motivation, engagement, collaboration, and content learning. We also emphasize that most of the works are developed in academic environments (with undergraduate students), others in practical environments (with industry professionals). Generally, the types of tests investigated are different, sometimes a more specific type of test or in a more general context, such as defect detection only. But, it is not always possible to identify in which development process the work was applied or which development methodology was adopted.

8 CONCLUSIONS

This work investigates the use of the PBL and JiTT methodologies to teaching ET for a DSD team. Based on a literature review and evaluation of the resources available, we planned and performed a training course on ET and analyzed the results obtained. While teaching ST has been challenging, under the circumstances imposed by social distance, where each team member is working remotely and isolated, teaching such a subject becomes even more challenging.

The use of these methodologies significantly contributed to the success of the course. They provided the grounds for adopting a real problem, assessing the students' needs with resources available before the class, adjusting the course to meet students' expectations and needs, and promoting collaboration. Additionally, the existence of a support tool for ET was key to optimize remote learning.

As further work, we intend to (1) monitor the execution of ET during the participants' daily agile development and then analyze the real advantages and difficulties experienced in this context, to (2) propose an approach that facilitates the implementation of ET, considering the DSD scenario, the generation of simple and robust ET artifacts. Additionally, we intend (3) to reuse this teaching experience in undergraduate ST courses.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments. The second and third authors were supported by CNPq/Brazil (processes 315057/2018-1, 437029/2018-2, and 311215/2020-3).

REFERENCES

[1] Agile Alliance. 2016. Agile glossary. URL: https://www.agilealliance.org/agile101/agile-glossary/ Accessed on August 13, 2020.

[2] Maurício Aniche, Felienne Hermans, and Arie van Deursen. 2019. Pragmatic Software Testing Education. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. ACM, New York, NY, USA, 414–420.

[3] James Bach. 2003. Exploratory testing explained.

[4] Charles C Bonwell and James A Eison. 1991. *Active Learning: Creating Excitement in the Classroom. 1991 ASHE-ERIC Higher Education Reports.* ERIC, sl.

[5] Ana Karina Silva de Castro. 2018. *Testes Exploratórios: Características, Problemas e Soluções.* B.S. thesis. Universidade Federal do Rio Grande do Norte.

[6] Jean Felipe P. Cheiran, Elder de M. Rodrigues, Ewerson Luiz de S. Carvalho, and João Pablo S. da Silva. 2017. Problem-Based Learning to Align Theory and Practice in Software Testing Teaching. In *Proceedings of the 31st Brazilian Symposium on Software Engineering* (Fortaleza, CE, Brazil) *(SBES'17)*. ACM, New York, NY, USA, 328–337.

[7] I Costa, S Oliveira, L Cardoso, A Ramos, and R Sousa. 2019. Uma Gamificação Para Ensino e Aprendizagem de Teste Exploratório de Software: Aplicação em um Estudo Experimental. *XVIII Simpósio Brasileiro de Jogos e Entretenimento Digital (Education Track–Short Papers)* 2019, 1 (2019), 1232–1235.

[8] Emanuel Ferreira Coutinho and Carla Ilane Moreira Bezerra. 2018. Uma avaliação inicial do jogo para o ensino de testes de software iTestLeaening sob a ótica de um software educativo. In *Congresso sobre Tecnologias na Educação*, Vol. 3. SBC Open Library, Fortaleza,CE, 11–22.

[9] Lisa Crispin and Janet Gregory. 2009. *Agile testing: A practical guide for testers and agile teams.* Pearson Education, sl.

[10] Stevão Alves de Andrade, Vânia de Oliveira Neves, and Márcio Eduardo Delamaro. 2019. Software Testing Education: Dreams and Challenges When Bringing Academia and Industry Closer Together. In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering* (Salvador, Brazil) *(SBES 2019)*. ACM, New York, NY, USA, 47–56.

[11] Igor Ernesto Ferreira Costa and Sandro Ronaldo Bezerra Oliveira. 2020. The use of gamification to support the teaching-learning of software exploratory testing: an experience report based on the application of a framework. In *2020 IEEE Frontiers in Education Conference (FIE)*. IEEE, Uppsala, Sweden, 1–9.

[12] Caliane de O Figuerêdo, Simone C dos Santos, PH Borba, and GH Alexandre. 2011. Using PBL to develop software test engineers. In *International Conference on Computers and Advanced Technology in Education.* sn, Cambridge, United Kingdom, 305–322.

[13] Vahid Garousi, Michael Felderer, Marco Kuhrmann, and Kadir Herkiloğlu. 2017. What Industry Wants from Academia in Software Testing? Hearing Practitioners' Opinions. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering* (Karlskrona, Sweden) *(EASE'17)*. ACM, New York, NY, USA, 65–69.

[14] Vahid Garousi, Austen Rainer, Per Lauvås, and Andrea Arcuri. 2020. Software-testing education: A systematic literature mapping. *Journal of Systems and Software* 165 (2020), 110570.

[15] Ahmad Nauman Ghazi. 2017. *Structuring Exploratory Testing Through Test Charter Design and Decision Support.* Ph.D. Dissertation. Blekinge Tekniska Högskola.

[16] Elisabeth Hendrickson. 2013. *Explore it!: reduce risk and increase confidence with exploratory testing.* Pragmatic Bookshelf, sl.

[17] Felipe Torres Leite, Jarbele C. S. Coutinho, and Reudismam Rolim de Sousa. 2020. An Experience Report About Challenges of Software Engineering as a Second Cycle Course. In *Proceedings of the 34th Brazilian Symposium on Software Engineering* (Natal, Brazil) *(SBES '20)*. ACM, New York, NY, USA, 824–833.

[18] Alexandra Martinez. 2018. Use of JiTT in a graduate software testing course: an experience report. In *2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET).* IEEE, Gothenburg,Sweden, 108–115.

[19] Jeffrey J McConnell. 1996. Active learning and its use in computer science. In *Proceedings of the 1st Conference on integrating Technology into Computer Science Education.* ACM, Barcelona,Spain, 52–54.

[20] Gregor M Novak. 2011. Just-in-time teaching. *New directions for teaching and learning* 2011, 128 (2011), 63–73.

[21] Marlla Rúbya F. Paiva, José Reginaldo F. Parente, Israel Rocha Brandão, and Ana Helena B. Queiroz. 2016. Metodologias ativas de ensino-aprendizagem: revisão integrativa. *SANARE-Revista de Políticas Públicas* 15, 2 (2016), 145–153.

[22] Leo Natan Paschoal and Simone do Rocio Senger de Souza. 2018. A Survey on Software Testing Education in Brazil. In *Proceedings of the 17th Brazilian Symposium on Software Quality* (Curitiba, Brazil) *(SBQS)*. ACM, New York, NY, USA, 334–343.

[23] Leo Natan Paschoal, Laíza Silva, and Simone Souza. 2017. Abordagem flipped classroom em comparação com o modelo tradicional de ensino: uma investigação empírica no âmbito de teste de software. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE).* SBC Open Library, Recife,PE, 476.

[24] Leo Natan Paschoal and Simone RS Souza. 2018. Planejamento e aplicação de flipped classroom para o ensino de teste de software. *RENOTE* 16, 2 (2018), 606–614.

[25] Rafael Queiroz, Fabrício Pinto, and Paulo Silva. 2019. IslandTest: jogo educativo para apoiar o processo ensino-aprendizagem de testes de software. In *Anais do XXVII Workshop sobre Educação em Computação.* SBC Open Library, Belém,PA, 533–542.

[26] Paula Raappana, Soili Saukkoriipi, Ilkka Tervonen, and Mika V Mäntylä. 2016. The Effect of Team Exploratory Testing–Experience Report from F-Secure. In *2016 IEEE Ninth International Conference on Software Testing, Verification and Validation Workshops (ICSTW).* IEEE, Chicago, IL, USA, 295–304.

[27] Lilian Passos Scatalon, Jeffrey C. Carver, Rogério Eduardo Garcia, and Ellen Francine Barbosa. 2019. Software Testing in Introductory Programming Courses: A Systematic Mapping Study. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19).* ACM, New York, NY, USA, 421–427.

[28] Beni Suranto. 2015. Exploratory software testing in agile project. In *2015 International Conference on Computer, Communications, and Control Technology (I4CT).* IEEE, Kuching, Malaysia, 280–283.

[29] James A Whittaker. 2009. *Exploratory software testing: tips, tricks, tours, and techniques to guide test design.* Pearson Education, sl.

[30] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering.* Springer Science & Business Media, sl.