Development of a teaching plan to support learning activities of exploratory test design and execution

Igor Ernesto Ferreira Costa
Graduate Program in Computer Science (PPGCC), Institute of Exact and Natural Sciences (ICEN)
Federal University of Pará (UFPA)
Belém – PA, Brazil
igor.ef.costa@gmail.com

Sandro Ronaldo Bezerra Oliveira
Graduate Program in Computer Science (PPGCC), Institute of Exact and Natural Sciences (ICEN)
Federal University of Pará (UFPA)
Belém – PA, Brazil
srbo@ufpa.br

2022 IEEE Frontiers in Education Conference (FIE) | 978-1-6654-6244-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/FIE56618.2022.9962486

***Abstract*— This Research Full Paper presents that nowadays it can see the growing evolution of the software development process, making agile methodologies commonly used. In this context, exploratory testing stands out as an alternative used in the industry to meet the needs of agile and / or short-term testing processes. However, professionals use this agile testing approach in an unstructured way, and this may be because they don't really understand how to apply it strategically where they can involve design and execution activities. In this context, this paper aims to present an approach that allows to apply the teaching of design and execution activities of exploratory test following structured procedures and aligned to the curricula (Brazilian and international) of the software engineering together with relevant practices in the industry. This study originates from a literature review, where important gaps are perceived to address exploratory testing education regarding design and execution activities. This involves the need to map the assets present in the curricula of the Brazilian Computer Society (SBC) and the Association for Machinery and Institute of Electrical and Electronics Engineer (ACM / IEEE), as well as being guided by the practices prescribed in the TMMi, by have a process area dedicated to these activities. In addition, we highlight the importance of having conducted interviews with professionals to identify tools, techniques and work products commonly used in the industry so that all this supports the alignment between academic and industrial context. In this way, it was possible to develop a consistent program of studies that allowed the construction of a teaching plan involving active pedagogical practices so that students can "learn by doing" in a more interactive way, that is, provide a student-centered teaching and learning in order to make such an approach more beneficial to the student, conditioning them to obtain the competences and abilities expected in the industry.**

Keywords*—exploratory testing, test design and execution, software testing education, active methodologies.

I. INTRODUCTION

Currently, agile development approaches have become essential, and it is also necessary to guarantee the quality of the products and services offered. Consequently, new testing approaches, considered agile in the specialized literature, have been the protagonist in the industrial scenario [1]. In this case, exploratory testing (ET) has been widespread to meet these agile quality processes [2]. However, many professionals (software engineers) understand how an informal testing approach, without any structured procedures, does not support the management of this testing process [3].

Despite the rise in the use of agile tests, especially exploratory testing, great challenges are still perceived due to the misunderstanding of how to more systematically apply design activities, exploration techniques to guide systematic navigation through the system under test and use exploratory test management, which provide structured procedures to assist incident tracking, control, metrics, and test process management [4]. In this context, it was observed the fact that the author of this work detected in a literature review that most of the studies deal only with the execution of exploratory tests, so there is a need to apply this study aiming to provide graduates not only with an understanding of the possibility of structuring exploratory test design and execution activities, but mainly providing them with skills that meet the demand of the industry [5].

It is extremely important that professionals are able to understand how to structure their testing processes and can have a more interactive and communicative contact with other employees, however, this must be acquired as part of their training. In this context, the proper learning process is essential, including for understanding that an industrial process with systematic procedures based on formal guidelines, standards or good practice guides makes it possible to achieve a significant level of effectiveness, as they are documents containing organized records of experienced professionals, uniting theory and practice [6] [7].

Conducting these studies on exploratory testing education becomes important as it involves theoretical and practical activities that are more in line with reality, aiming to train professionals with sufficient basic capacity to engage in the industry. Therefore, this study aims to present a teaching plan that provides the practical application of software testing concepts, in particular, activities aimed at the design and execution of software testing, in line with the Test Maturity Model integration (TMMi), and the SBC and ACM / IEEE curricula. This teaching and learning plan becomes appropriate and effective related to Exploratory Test Design and Execution, including using alternative pedagogical practices to traditional teaching, strongly defended in the Computing Curricula (CC) 2020 [8][9].

In addition to this introductory section, the paper is structured having in Section II the theoretical foundation, in Section III the related works, in Section IV the research methodology, in Section V the study program adapted for this subject of exploratory test design and execution, in Section VI the teaching plan based on the syllabus, in Section VII the prior evaluation of this teaching plan and, finally, in Section VIII the conclusion and future work.

II. BACKGROUND

This section presents the concepts of software engineering education, the exploratory testing approach and a description of the Test Design and Execution process area prescribed in TMMi.

A. Software Testing Education

Software Engineering Education (SEE) continues to evolve, with the main focus being the preparation of Software Engineering (SE) students for future careers, however there are still large gaps between the subjects covered in academia (university) with techniques and models (frameworks) used by the software industry. Such gaps are caused, mainly, in subjects that have been spread exponentially, for example, quality assurance, machine learning, as well as technological innovations [10].

For this, international organizations such as the ACM and IEEE; and national, such as the Brazilian Ministry of Education (MEC) and the SBC guide the SE curricula considering the integration of industrial perspectives and alternative teaching approaches to the classical model. However, the nature of SE learning activities in traditional classrooms is limited in scope and time, making it more difficult to strike a proper balance between theory and practice and meet industrial demands becoming an open question for Software Engineering Education [10].

Among the corresponding teaching guidance documents published by the aforementioned organizations, it is emphasized that the latest version of the computing curriculum (CC 2020) which strongly supports the use of a teaching and learning approach that offers practical experiences adherent to the industry and that promotes the greater student engagement, considering the teacher a facilitator of this process [9]. Therefore, this work presents the documentary inputs used as a basis for identifying topics (asset mapping) related to the context of this research.

B. Exploratory Testing

It is defined as an approach that is based on the tester's experience, where the design and execution of tests are carried out simultaneously and freely, following the knowledge acquired from each previously performed exploration and analysis of the results of previous tests. In this way, the tester is able to identify heuristics about the behavior of the system under test and types of failures already identified in the system under test or by familiarity with a system in the same application domain [11][12][13][14].

In [15], exploratory testing is classified as an agile testing approach that can be applied according to what is proposed by Brian Marick in the Agile Testing Quadrant, presented in Figure 1 [16].

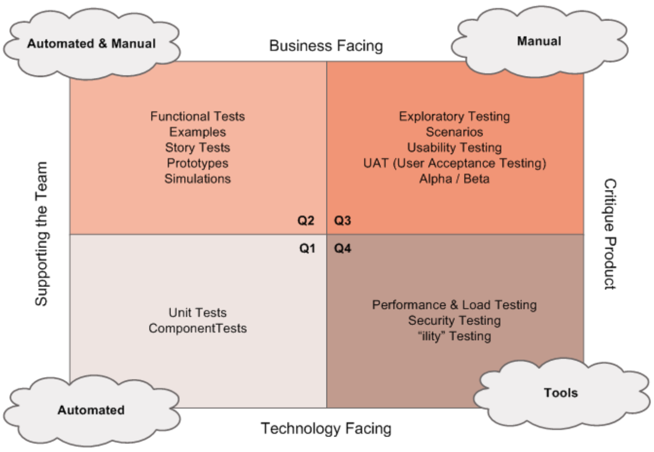

Fig. 1. Agile Testing Quadrants [15][16].

In [17], a reformulation of the concept of exploratory testing is presented, defining an approach that allows evaluating a product by learning about it through exploration and experimentation, involving to some degree: questioning, study, modeling, observation, inference, etc. [17]. Towards this end, authors in[18] identified in their studies that testers implicitly apply many exploration strategies depending on the level of education.

In [19], it is exemplified that the learning process using the exploratory test approach occurs in a cyclical way. This model is called the Kolb Learning Cycle (as can be seen in Figure 2), where the tester holds experiences, then makes inferences and abstracts what he thinks is relevant and what he managed to perceive about the test object, thus actively experiencing the object. In this context, [20] presented results that show that this Experimental Learning was essential to guide or organize the thinking of testers to detect defects.

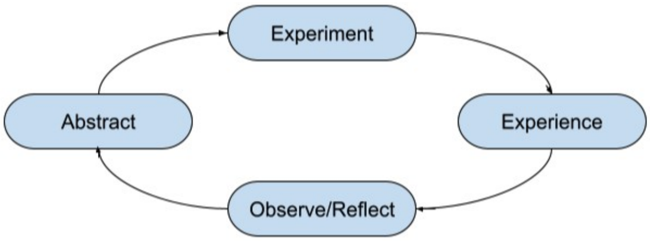

Fig. 2. Kolb Learning Cycle [19].

In view of this, [19] demonstrates how the dynamics of the exploratory test process involving Kolb's Experimental Learning can be structured (as can be seen in Figure 3). This is constituted from the elaboration of the test chart, proceeding with the identification of useful variables and heuristics to observe ways and/or data to explore the object under test. With this, it allows to conduct the experiment of the system (exploration) being carried out careful observations as this exploration of the object takes place. Finally, an interrogation is carried out with the technical leader in order to understand the level of knowledge acquired and/or identify the test coverage explored, and identify opportunities for improvement, etc.

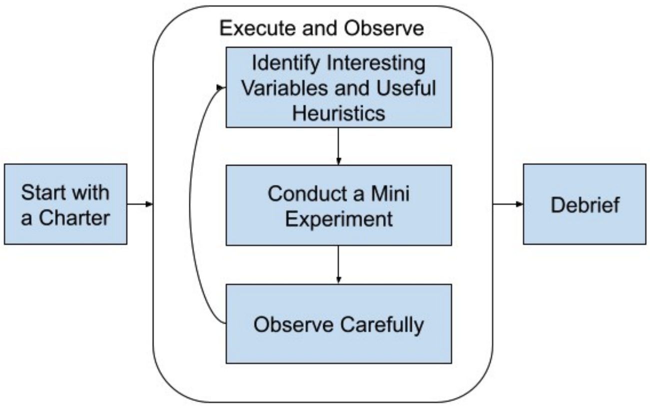

Fig. 3. Structured exploratory testing process involving Kolb's Experimental Learning [19].

It is also emphasized the possibility of using exploration strategies (navigation by the test object) to make the tests more efficient [18][19][21]. In addition, [4] states that because there are some deficiencies that directly affect the management of test processes, techniques have emerged to minimize it, such as Session-Based Test Management (SBTM), Thread-Based Test Management, Risk-Based Test Management. In this case, such management techniques define more systematic procedures to provide a structured application of exploratory testing, managing to meet factors relevant to the effectiveness of the testing process.

C. Test Design and Execution Prescribed at TMMi

In TMMi there is a process area that specifically serves the Test Design and Execution activities, which aims to improve the capacity of the test process regarding the execution of architecture construction activities (design), test execution and analysis from the definition of design technique specifications [22][23].

In TMMi are proposed objectives, practices and their corresponding sub-practices to be carried out in a structured test process, encompassing from the identification of test conditions to the management of incidents to closure. For this, test design techniques are reported to derive test conditions and test procedures, exemplifying the use of test charts for exploratory testing [22][23].

In addition, in order to achieve the effectiveness and efficiency of the test process, the possibility of using tools both to assist these test design activities (identify test conditions, procedures and data, maintain bidirectional traceability and prepare a test schedule) regarding execution-oriented practices, considering process and incident management (reporting incidents, communicating incidents to stakeholders, evaluating incidents for closure, regression testing and managing incidents) [22][23].

D. Active Methodologies

Active methodologies have gained notoriety and more importance in the last decade, although there are records that encourage such teaching for many years. This exponential growth in the search for new pedagogical practices occurs, mainly, from the insertion of computers and access to the Internet as an educational resource, so there is a need to obtain teaching alternatives to improve learning, generally following the direction of use of more engaging strategies than traditional teaching [24]. It is considered that the importance of student-centered pedagogical practices emerged more in debates since the last century with the movement called "New School", in which some thinkers of the time, for example, William James and John Dewey, already defended a methodology of teaching with a focus on learning through experiences and autonomous student development [25].

It is emphasized that the idea of the New School, preached by John Dewey, is based on "learning by doing", in experiences with educational potential, which is notable in active methodologies. According to [25], the active methodology is characterized by the interrelationship between education, culture, society, politics and school, where it develops through active and creative methods centered on the learner's activity in order to promote learning. For [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 as a guide for this process. In this context, [26] exposes aspects that can characterize the role of the teacher and the student (as can be seen in Table I).

TABLE I. TEACHER VS STUDENT [26].

Teacher	Student
Doesn't speak, *ask*.	Don't take notes, *look, find.*
Suggests topics and instruments	Search and find solutions
Learns technology with students	Learn about quality and rigor with the teacher
Evaluates student solutions and responses, examining quality and accuracy; contextualization.	Refines and improves responses, adding rigor, context, quality.

Finally, it is noteworthy that there are several techniques or methods associated with active methodology, which provide the student with learning through experiences that boost their development of autonomy and protagonism, as well as create situations that arouse their curiosity and become aware of reality. Such methods are: flipped classroom [25][27], project-based learning [28], problem-based learning [28][29], shared classroom [30], blended learning [26][30], game-based learning (gamification) [25] [39], etc. It is emphasized that the teacher has the autonomy to adapt or create methods, while respecting the principles of active methodology.

III. RELATED WORKS

In [31], there was the elaboration of a more generic teaching plan returning to the teaching and learning of agile software testing foundations using active methodologies. This teaching plan was based on an expert literature review and asset mapping. As a result, it is possible to observe a close alignment with the ACM / IEEE and SBC curricula and guides such as TMMi and SWEBOK, use of many active pedagogical practices, being designed for face-to-face teaching, but it is not described if the test approaches to be taught must be structured. This present study differs in that it does not use SWEBOK, as it focuses on practical Test Design and Execution activities, instead of presenting different concepts about the same term and technique.

In [32], a pragmatic software testing subject was proposed based on nine key principles that can be taught to computer science students including building a testing mindset and interacting with practitioners. This Software Testing and Quality Engineering subject covers several different aspects of software testing in general and adheres to the syllabus of the International Software Testing Qualifications Board (ISTQB). It is noticed that the initial syllabus from the test foundation, presenting different techniques and ending with lectures by professionals working in the area. It is noted that this present work differs in being based on the Brazilian and international curricula, following the focus on exploratory test design and execution activities adhering to the TMMi guide, which also involves programmatic content based on the ISTQB syllabus [33].

Therefore, it is clear some similarities with the other works when observing the documents analyzed to carry out the mapping of assets and some steps for the construction of its proposal (teaching-learning approach). However, the differential of this work is due to its focus on the practical application of Exploratory Test Design and Execution adhering to the academic and industrial context, being planned to meet the need for remote teaching caused by the current pandemic moment. In addition, it is observed that the pedagogical practices used are grounded and consistent with teaching and learning using digital technologies.

IV. RESEARCH METHODOLOGY

The methodological procedures applied were:

i) **Definition of the study theme**, a literature review was carried out and a great potential for study on software engineering education was identified, more specifically on Exploratory Test Design and Execution [5]. In this case, the relevance of carrying out a documentary analysis of curricular guidelines was perceived, as well as investigating tools, techniques and work products used in the industry to support the construction of a teaching plan, offering practical applications relevant to the industry and adhering to academic curricula.

ii) **Mapping of assets**, the analyzed inputs were the RF-SBC, CS-Curricula of ACM / IEEE and TMMi. The first two were selected because they are Brazilian and international curriculum guides, respectively, and the last one because it is a guide based on practical and structured applications of Test Design and Execution. In view of the inputs, a mapping of assets was carried out from the identification and cross analysis of data present in the curricular guidelines with the TMMi referring to the assets relevant to the Test Design and Execution activities, in general [34]. In this mapping, 13 assets and 110 asset items for Test Design and Execution were identified, involved in a two-level correspondence, namely: a) Training axes (RF-SBC) and related knowledge areas (ACM / IEEE) the Test Design and Execution process area, b) Derived Content and Competences (RF-SBC), as well as topics and learning outcomes (ACM / IEEE), which were related to the specific objectives, specific practices and sub-practices of the TMMi process area, which is the focus of this work [34].

iii) **Interviews with professionals**, the participants were professionals in the software testing accredited by a national (Brazilian) and / or international institution, or obtained professional certification at TMMi, including experience in test process improvement. Thus, it was possible to obtain relevant answers to the construction of a teaching plan focused on the Design and Execution of Exploratory Test to offer practical content closer to reality [35]. The results of the interviews were: a) identification of tools, techniques and work products for Exploratory Test Design and, b) identification of tools, techniques and work products for Exploratory Test Execution. In the context of item “a”, the most used tools to assist design activities were Testlink and Jira, where risk analysis was the most cited activity in relation to support to identify and prioritize test conditions, including being useful as well as complement to the Exploratory Test. Regarding work products, the test plan and results of previous test runs were most used. In the context of item “b”, Mantis and Jira were the most mentioned tools to help the management and execution of tests. As for the execution techniques, the use of Exploratory Testing with manual and automated strategy was observed, while the Incident Report and the Matrix were commonly used work products [35].

iv) **Construction of the Study Program**, based on the mapping, 12 competences expected from the Exploratory Test Design and Execution were identified, where it was possible to establish correlated subjects organized in four teaching units that comprise: a) Exploratory test analysis and design, b) Implementation of exploratory testing procedures, c) Exploratory test execution and d) Test and incident process management. Subsequently, the structural elements for each teaching unit were defined, which are: prerequisites, guiding questions, programmatic contents, expected results and learning levels, also having the teaching strategy that corresponds to the approach to be used in the teaching plan, being adhering to the learning levels established by the teaching unit [36].

v) **Application of peer review**, this was applied to the mapping of assets, analysis of interview questions, study program and teaching plan to ensure from the beginning the adherence of the analyzed inputs until the elaboration of the proposed teaching plan based on active methodologies. In summary, it was a constant activity at each stage of this research, in order to also avoid author bias.

V. Syllabus

The teaching plan proposed in this work is an instance of the syllabus already developed, which aims to support the teaching and learning of exploratory test design and execution in undergraduate courses in Computer Science [36]. It is emphasized that this syllabus was prepared from a mapping of assets based on ACM / IEEEE (international) and SBC (Brazilian) curricula, and the TMMi (international practical guide), in addition to the results obtained in interviews with professionals to coherently align theory (academic context) with practices commonly used in the industry. Therefore, Table II presents the adopted competences that students should obtain when studying the subject of exploratory test design and execution.

TABLE II. General Competence Adopted

C1. Employ methodologies that aim to ensure quality criteria throughout the exploratory test design and execution step for a computational solution.
C2. Apply software maintenance and evolution techniques and procedures using the ET approach.
C3. Manage the exploratory test approach involving basic management aspects (scope, time, quality, communication, risks, people, integration, stakeholders and business value).
C4. Apply techniques for structuring application domains characteristics in the exploratory test approach.
C5. 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.
C6. Apply software model analysis techniques to enable traceability of test conditions and test data (with a focus on exploratory testing) to requirements and work products.
C7. Apply theories, models and techniques to design, develop, implement and document exploratory testing for software solutions.
C8. Apply validation and verification techniques and procedures (static and dynamic) using exploratory testing.
CG9. Preemptively detect software failures on systems from the exploratory test application.
C10. Perform integrative testing and analysis of software components using ET in collaboration with customers.
C11. Conduct exploratory testing using appropriate testing tools focused on the desirable quality attributes specified by the quality assurance team and the customer.
C12. Plan and drive the process for designing test cases (charters) for an organization using the ET approach.

The syllabus used in this study, adapted from [36], follows the structural organization (see in Table III):

- **Prerequisites** are the subjects previously necessary to provide the theoretical basis on the area,
- **Guiding Questions** are inquiries made to students, usually at the beginning of the teaching unit to encourage discussion on the subject,
- **Programmatic Contents** are the topics to be taught in line with the competences planned for the subject,
- **Teaching Strategy** is the approach adopted to teach the syllabus, being established in accordance with the desired learning level and expected results,

- **Expected Results** are the expected competences and skills that the student will be able to obtain in order to retain the knowledge acquired with the subject,
- **Learning Level**, for each expected result there is an expected learning level and for this, the Revised Bloom Taxonomy [37][38] was used.

TABLE III. SYLLABUS FOR EXPLORATORY TESTING DESIGN AND EXECUTION

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? (PC1 - 1.1, 1.2) Q2. What criteria are suitable for prioritizing test conditions and test procedures, both exploratory? (PC1 - 1.2) Q3. What aspects are analyzed for the strategic definition of an exploratory testing process? (PC1 - 1.3) Q4. How to develop test charts suitable for certain contexts? (CP2 - 2.1) Q5. How to identify suitable criteria for verification of initial tests? (CP2 - 2.1) 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? (CP4 - 4.1) Q8. How to use good practices to communicate incidents to stakeholders? (CP4 - 4.1)	
Programmatic Content – PC 1.1	
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	***Learning Level***
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 procedures.	Remember / Factual Understand / Conceptual
The student must be able to identify the test coverage to be achieved with test design techniques in line 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	
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	***Learning Level***
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	
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	***Learning Level***
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 SBTM.	Understand / Conceptual Apply / Conceptual and Procedural Analyze / Procedural
The student should be able to understand and apply good practices in incident logging and cause analysis.	Remember / Factual Understand / Conceptual Apply / Procedural Analyze / Procedural
Programmatic Content – PC 1.4	
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	***Learning Level***
The student must be able to understand and apply good practice in reviewing reports and writing a test summary report.	Understand / Conceptual Apply / Conceptual and 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

VI. TEACHING PLAN

From the study program, this teaching plan was defined, where the learning strategies were established, the description of the dynamics of the activities (methodological procedure) and the expected learning level for each of these pedagogical techniques used (active methodologies). It is emphasized that this teaching plan focuses on a student-centered approach, where the teacher becomes a facilitator of this process, as an alternative to breaking the paradigm of traditional teaching with massive lectures.

In this context, some gamification elements were used as a way of trying to provide an even greater interaction between students. For this, it is possible to observe such elements based on the Octalysis Framework established by [39] and its purpose in the context of this work.

Mystery box: it gives awards (virtual symbols of achievement) when the team or student achieves an excellent performance in the delivery of the activity. The criteria to be analyzed: i) organization of work, ii) expected result (objective), iii) on-time delivery and iv) team involvement (attitude).

Free lunch: it gives a prize (gift) to the team when they reach the goals, as a way of rewarding the good result of the work only at the end of each general teaching topic.

Achievement Symbol: these are virtual medals to be awarded, when necessary, to the student or team.

Progress bar: throughout the study, the evolution progress of the teams must be available to all participants, specifically in relation to the project in question being developed. Progress tracking should be viewed in the form of a chart on the online wall or similar instrument.

sounds a lot like gamification

Build from scratch: this element is inherent to activities that need to be solved collaboratively and cooperatively by each participant, who must contribute to achieving the objective in common agreement between all.

Appointment dynamics: it is used to observe the responsibility established for each team member. In this case, each team must have a Kanban board that must be visible only to the team members and teacher during the entire execution of the developed project.

Group quests: the purpose of this element is for all team members to contribute to achieving the objective of the activity, working to meet the analysis criteria with an average considered at least good.

Faced with the moment of a pandemic experienced, where there are still uncertainties about when to return and what are the appropriate procedures to be adopted to return to face-to-face teaching, 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 to the use of active methodologies through digital technologies.

The EOL educational modality [40] 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 of the teaching-learning process different from conventional Distance Education (DE). EOL is characterized by eight principles: a) knowledge as an “open work”, b) content curation, synthesis and study guides, c) Diverse computing environments, d) networked, collaborative learning, e) conversation among all, in interactivity, f) authorial activities inspired by cyberculture practices, g) online teaching mediation for collaboration and h) formative and collaborative evaluation, based on competences.

A. *Exploratory Test Analysis and Design*

For this programmatic content, collaborative dialoged expository classes will be taught, using the basic concepts of testing levels, types and techniques to provide a holistic view of Software Testing and allow students to perform a more accurate identification of which moment in the software life cycle the Exploratory Testing approach may be applied. In addition, exploratory test management (SBTM) techniques, test design techniques and work products will be exposed for students to analyze and, consequently, support them in the elaboration of charters aligned with the test objective.

These techniques must be approached with fixation exercises in a practical way, obeying the learning levels and expected results for this topic. A practical activity of deriving and prioritizing test conditions must be carried out, identifying test data and procedures, having the need to document them, including generating a matrix to maintain the traceability of these generated work products. Regarding prioritization, it can be performed using the COORG technique (Acronym for: Classify, Order and Organize) inherent to the Product Backlog Building (PBB) [41].

Beforehand, it is emphasized that these activities, as well as for the other topics, must be taught synchronously, where a practical project must also be produced putting into practice the concepts taught and already briefly applied in these synchronous classes. In this case, the practical project must be applied asynchronously, however monitored by instant conversation tools and support for establishing responsibility on the Kanban board. Another point is that the teacher will have the role of facilitator during the synchronous moment, and will act as a client when it is time to develop the practical project, establishing possible objectives and goals for delivery, and this practical project must be presented in the form of a flipped classroom at the end of the study of a programmatic content.

Therefore, the structure of the teaching plan for this programmatic content can be seen in Table IV.

TABLE IV. STRUCTURE OF TEACHING PLAN FOR PC 1.1

Competences in Table II	**Software Tools**
C1, C4, C5, C6, C7, C12	Google Meet, Email, App of Instant Chat, Trello, Jira, TestLink and Microsoft Office.
Active Technique	Dialogued Class with Collaborative Practice, Problem Based Learning, Team Based Learning, Flipped classroom.
Evaluation	Fixation Exercises (practical activity), Practical Project.
Reference Material	
- More Agile Testing: Learning Joureys for the Whole and Team [1] - Session-Based Test Management [4] - ISO/IEC/IEEE 29119-1 [11] - Explore it!: Reduce Risk and Increase Confidence with Exploratory Testing [19] - TMMi in the Agile world [23] - Foundation Level Syllabus [33] - Product Backlog Building (PBB) [41] - The Art of Software Testing [42] - Exploratory Testing [43] - TestLink User Manual [44]	

B. *Implementation of Exploratory Test Procedures*

For this programmatic content, examples of charters will be presented to support the elaboration of new charters through the collaboration of all. This methodology can also be applied to the activity of defining verification criteria for initial tests on the main functionalities of the system (intake testing). In this case, an activity divided by group should be applied so that there can be a more direct collaboration and interaction between the students.

The concepts and examples of exploration techniques are important to correlate them with the pre-defined charters, as well as to jointly analyze some relevant aspects in defining a test schedule. The application of peer learning will involve the knowledge seen in the two topics, being for the elaboration of charters, where it must be established which exploration techniques will be used, and to establish an adequate execution schedule for these charters. It is noteworthy that the practical project also occurs with the same dynamics described for Exploratory Test Analysis and Design.

Therefore, the structure of the teaching plan for this programmatic content can be seen in Table V.

TABLE V. STRUCTURE OF TEACHING PLAN FOR PC 1.2

Competences in Table II	Software Tools
C5, C7, C8, C10, C12	Google Meet, Email App of Instant Chat, Trello, Jira, TestLink, Microsoft Office.
Active Technique	Dialogued Class with Collaborative Practice, Team Based Learning, Flipped classroom.
Evaluation	Fixation Exercises (practical activity), Practical Project.
Reference Material	
- More Agile Testing: Learning Joureys for the Whole and Team [1] - Session-Based Test Management [4] - ISO/IEC/IEEE 29119-1 [11] - Explore it!: Reduce Risk and Increase Confidence with Exploratory Testing [19] - Exploratory Software Testing: Tips, Tricks, Tour, and Techniques to Guide Test Design [21] - TMMi in the Agile world [23] - Foundation Level Syllabus [33] - Product Backlog Building (PBB) [41] - Exploratory Testing [43] - TestLink User Manual [44]	

C. *Exploratory Test Execution*

For this programmatic content, some practical activities will be applied to the exploratory test in a collaborative way, being dynamized based on Dojo Randori and Dojo Kake (LAB) [45]. It is noteworthy that at first there will be an activity in the classroom and, later, a group activity, being extraclass. In the classroom, there will be basic activities for the assertive understanding of the use of exploration techniques and in the extraclass activity it will involve the use of these pre-established exploration techniques, the use of a technique that allows the structuring of these activities related to execution (example, SBTM), as well as running the tests according to the schedule.

The dialogued classes with collaborative learning will serve to pass on the good practices of writing a log or incident record, incident analysis establishing possible causes of incidents to eligible stakeholders, always maintaining traceability between test conditions, test procedures and test results. In addition, there must be group practice so that students can practice even more all these procedures together in extra-class moments (asynchronous moment). It is noteworthy that the practical project also occurs with the same dynamics described for Exploratory Test Analysis and Design.

Therefore, the structure of the teaching plan for this programmatic content can be seen in Table VI.

D. *Test and Incident Process Management*

This programmatic content should present good practices in relation to the review of incident reports. Subsequently, apply a paired incident log report review activity using the reports generated when searching for incidents (Exploratory Test Execution). In addition, practice how to write a test summary report following good practices for effective communication with stakeholders and understanding possible appropriate incident remediation actions as well as directing them to closure.

In addition, a dialogued class will also be applied with collaborative practical activity in which opportunities for improvement of some test process flows presented will be discussed. In the second moment, in a synchronous class, the groups should analyze their process together with the procedures and activities performed so far to carry out changes to improve this process, and make it more appropriate or viable (considering good practices) to execute it according to lessons learned.

TABLE VI. STRUCTURE OF TEACHING PLAN FOR PC 1.3

Competences in Table II	Software Tools
C2, C8, C9, C10, C11	Google Meet, Email App of Instant Chat, Trello, Jira, TestLink, Microsoft Office.
Active Technique	Dialogued Class with Collaborative Practice, Dojo Randori, Dojo Kake, Team Based Learning, Flipped classroom.
Evaluation	Fixation Exercises (practical activity), Dojo Randori, Dojo Kake, Practical Project.
Reference Material	
- Session-Based Test Management [4] - Explore it!: Reduce Risk and Increase Confidence with Exploratory Testing [19] - Exploratory Software Testing: Tips, Tricks, Tour, and Techniques to Guide Test Design [21] - TMMi in the Agile world [23] - Foundation Level Syllabus [33] - Exploratory Testing [43] - TestLink User Manual [44]	

Therefore, the structure of the teaching plan for this programmatic content can be seen in Table VII.

TABLE VII. STRUCTURE OF TEACHING PLAN FOR PC 1.4

Competences in Table II	Software Tools
C2, C3, C12	Google Meet, Email App of Instant Chat, Trello, Jira, TestLink, Microsoft Office.
Active Technique	Dialogued Class with Collaborative Practice, Team Based Learning, Flipped classroom.
Evaluation	Fixation Exercises (practical activity), Practical Project.
Reference Material	
- Session-Based Test Management [4] - Explore it!: Reduce Risk and Increase Confidence with Exploratory Testing [19] - TMMi in the Agile world [23] - Foundation Level Syllabus [33] - TestLink User Manual [44]	

As described for Exploratory Test Analysis and Design, hands-on design will also occur in this programmatic content with the same dynamics. It is worth mentioning that the software tools, techniques and work products used were listed based on the results of interviews with professionals [35][46]. Finally, it is pointed out that there will be specific feedback classes to collect qualitative data to be analyzed. These moments are important to identify strengths, weaknesses, opportunities and threats regarding software tools, material resources and applied activities, as well as to observe the satisfaction degree and self-evaluation of students about the learning obtained.

VII. EVALUATION

For prior evaluation of the teaching plan, it was submitted to Peer Review, in which the results are presented in Table VIII. For this, an identifier (ID) was assigned to each change request, a category to which the request belongs is defined, the item to be adjusted is listed, as well as the comment that justifies the reason for the adjustment and the suggestion for improvement. In this case, the categories are:

- **High Technician - TA**, indicates that a problem was found in an item that, if not changed, will compromise the considerations,
- **Low Technician - TB**, indicates that a problem was found in an item that it would be convenient to change,
- **Editorial - E**, indicates that a Portuguese error was found or that the text could be improved,
- **Questioning - Q**, indicates that there were doubts about the content of the considerations,
- **General - G**, indicates that the comment is general in terms of considerations.

TABLE VIII. ADJUSTMENT ITENS ON THE PEER REVIEW

ID	Category	Item
1	E	Teaching Strategy
Comment: There are words and grammatical and spelling errors throughout the text. **Suggestion:** Correct these errors and replace the words with more formal texts.		
2	E	Gamification Elements
Comment: There is no contextualization about the use of the active methodology of gamification. **Suggestion:** Clearly present that gamification was selected as an active methodology to use in the teaching strategy.		
3	TB	Competences
Comment: Competences are comprehensive for any type of test. **Suggestion:** Customize these competences for Exploratory Testing.		
4	TB	Subject Planning
Comment: It was not specified the instantiation of the pre-established syllabus for the referred subject, being adhering to the TMMi. **Suggestion:** Define that the programmatic contents was based on the generated syllabus and maintained compliance with the TMMi.		
5	TA	Selection of Pedagogical Practices
Comment: The selection of pedagogical practices with the use of digital technology established for the teaching strategy was not justified. **Suggestion:** Justify the selection of practices also based on the online teaching modality (EOL)		
6	E	Learning Level
Comment: There is no reference from where the Learning Level elements were extracted. **Suggestion:** Reference the Revised Bloom's Taxonomy.		
7	TB	Expected Results
Comment: The expected results described do not have an objective detail. **Suggestion:** Be more clear and objective in detailing the expected results.		
8	TB	Learning Level versus Expected Results
Comment: Some learning levels are not in line with expected results. **Suggestion:** Review the alignment between learning levels and expected results for each teaching unit.		
9	E	Software Tools
Comment: There is no reference from where the software tools for use in the course were extracted. **Suggestion:** Reference the work that supports the use of the selected software tools.		

All the adjustments requested by the experts were carried out by the author of this study, a fact that allowed the construction of a teaching plan that was strongly adherent to the analyzed curricula and the practices of the industry, to be achieved by the interviews with professionals and also by following the structure of activities prescribed in the TMMi.

VIII. CONCLUSION AND FUTURE WORK

The objective of this work was to present in detail the planned teaching plan from a previously constructed syllabus aimed at the practical application of structured Exploratory Test Design and Execution activities in line 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, even considering such an approach to remote teaching based on a teaching modality (EOL) that also proposes a more interactive, communicative and engaging classroom dynamics through the use of digital technologies, unlike the Distance Education modality.

It is emphasized that the adherence to the teaching plan follows from the documents obtained as a basis for analysis for mapping the assets, interviews with professionals in the area, alignment with the TMMi process area and the use of active methodologies through digital technologies, being evaluated its viability perennially by experts in the field, both in research on software engineering education and practical applications in the industry. However, it is intended to apply it in an undergraduate computer science class to validate its efficiency and effectiveness.

The validation of the approach is planned and it will be possible to obtain the results from the application of this teaching plan. Therefore, in the future, one should qualitatively evaluate with pre-test and post-test, feedback through SWOT (Strengths, Weaknesses, Opportunities, Threats) analysis, evaluation of satisfaction with the teaching approach, teamwork, communication, interaction, organization, deliveries, objective achieved in the activities, evaluation of activities regarding their alignment, coherence and feasibility according to established learning levels. In addition, evaluate quantitatively through the grades obtained by students in activities that are entirely practical.

In this way, the importance of this study is emphasized in providing the training of students at a basic level sufficient for use in the industry, even breaking the paradigm of understanding that exploratory testing is an approach only to be free, as well as to support new studies in the area.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] J. Gregory and L. Crispin, "More Agile Testing: Learning Joureys for the Whole and Team," Addison-Wesley Professional: USA. 2015.

[2] I. S. Elgrably and S. R. B. Oliveira, "The Importance of Application of Agile Tests in the Software Industry: An Exploratory Approach Using Interview," 14th International Conference on Information Systems & Technology Management – CONTECSI. DOI:10.5748/9788599693131-14CONTECSI/RF-4651. 2017.

[3] D. Pfahl, M. Mantila, H. Yin, and J. Much, "How is Exploratory Testing Used?: A state of the Practice Survey," ESEM'14, September 18-19, 2014, Torino, Italy. Copyright 2014 ACM 978-1-4503-2774-9/14/09.2014.

[4] J. Bach, "Session-Based Test Management," Software Testing and Quality Engineering Magazine. 2000.

[5] I. Costa and S. Oliveira, "An Evidence-Based Study on Automated Exploratory Testing," 17th CONTECSI. Brazil. 2020.

[6] A. Crespo, O. Silva, C. Borges, C. Salviano, M. Junior, and M. Jino, “Uma Metodologia para Teste de Software no Contexto da Melhoria de Processo,” In: SBQS. 2004.

[7] K. Naik and P. Tripathy. “Software Testing and Quality Assurance: Theory and Practice,” John Wiley & Sons, Inc. 2008.

[8] ACM/IEEE, “Computer Science Curricula 2013,” ACM and IEEE Computer Society, Incorporated: New York, NY, USA. 2013.

[9] ACM/IEEE, “Computing Curricula: Paradigms for Global Computing Education,” *ACM and IEEE Computer Society, Incorporated: New York, NY, USA*. 2020.

[10] O. Cico, L. Jaccheri, A. Nguyen-Duc, and H. Zhang, “Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends,” Journal of Systems and Software. Volume 172. 2020.

[11] ISO/IEC/IEEE 29119-1, “Software and systems engineering, Software testing, Part 1, Concepts and definitions,” ISO: Geneve. 2013.

[12] SWEBOK, “Guide to the Software Engineering Body of Knowledge V3.0,” IEEE Computer Society. 2014. Available in: http://www.computer.org/portal/web/swebok/.

[13] J. Bach, “Exploratory Testing. In: The Testing Software Engineer,” 2nd ed., E. van Veenendaal (Ed.) Den Bosch: UTN publisher, pp. 253-265. 2004.

[14] C. Kaner, “A Tutorial in Exploratory Testing,” QUEST 2008. Available on: <http://www.kaner.com/pdfs/QAIExploring.pdf>.

[15] J. Gregory and L. Crispin, “Agile Testing: A Pratictal Guide for Testers and Agile Teams,” Addison-Wesley Professional: USA. 2009.

[16] M. Huttermann, “Agile Record: The Magazine for Agile Developers and Agile Testers,” 2011. Available in: http://huettermann.net/perform/AgileALM-AgileRecord-Huettermann.pdf.

[17] J. Bach, “History of Definitions of ET,” SATISFICE:. Software Testing For Serious People. Available in: https://www.satisfice.com/blog/archives/1504. 2015.

[18] M. Micallef, C. Porter, and A. Borg, “Do Exploratory Testers Need Formal Training? An Investigation Using HCI Techniques,” IEEE. - Ninth International Conference on Software Testing, Verification, and Validation Workshops. 2016.

[19] E. Hendrickon, “Explore it!: Reduce Risk and Increase Confidence with Exploratory Testing,” Pragmatic Bookshelf; 1 edition. ISBN: 978-1937785024. 2013.

[20] B. Lorincz, B. Iudean, and A. Vescan, “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), August 23. 2021.

[21] J. Whittaker, “Exploratory Software Testing: Tips, Tricks, Tour, and Techniques to Guide Test Design,” 1st Ed. Pearson Education, Inc. 2010.

[22] E. Van Veenendaal, “Test Maturity Model integration – TMMi Release 1.0: Guidelines for Test Process Improvement,” Produced by TMMi Foundation. 2018.

[23] E. Van Veenendaal, “TMMi in the Agile world – TMMI Release 1.3,” Produced by TMMi Foundation. 2019.

[24] J. Mattar and W. Czeszak, “Avaliação em educação a distância,” In Faria, E.; Sousa, H.; Fernandes, T. (Orgs.). Educação a distância: textos aplicados a situações práticas. João Pessoa, Gráfica São Matheus. 2013.

[25] L. Bacich and J. Moran, “Metodologias Ativas para uma Educação Inovadora: Uma Abordagem Teórico-Prática,” Porto Alegre: Penso Editora. 2018.

[26] M. Prensky, “Teaching Digital Natives: Partnering for Real Learning,” Thousand Oaks, California. 2010.

[27] R. Talbert, “Guia para Utilização da Aprendizagem Invertida no Ensino superior – Flipped Learning na Universidade,” Porto Alegre, Editora Penso. 2019.

[28] W. N. Bender, “Aprendizagem Baseada em Projetos: Educação Diferenciada para o Século XXI,” Porto Alegre, Editora Penso. 2014.

[29] F. Camargo and T. Daros, “A Sala de Aula Inovadora: Estratégia Pedagógicas para Fomentar o Aprendizado Ativo,” Porto Alegre, Editora Penso. 2018.

[30] L. Bacich, A. T. Neto, and F. de M. Trevisani, “Ensino Hibrido: Personalização e Tecnologia na Educação,” Porto Alegre. Editora Penso. 2015.

[31] I. Elgrably and S. Oliveira, “Model for Teaching and Training software testing in an agile context,” In: FIE. 2020.

[32] M. Aniche, F. Hermans, and A. van Deursen, “Pragmatic Software Testing Education,” In SIGCSE ’19: 50th ACM Technical Symposium on Computer Science Education, February 27th - March 2, 2019, Minneapolis, MN, USA. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3287324. 3287461. 2019.

[33] ISTQB, “International Software Testing Qualifications Board - Foundation Level Syllabus,” 2018.

[34] I. Costa and S. Oliveira, “An Asset Mapping in the ACM / IEEE and SBC Curriculum to the Test Design and Execution of the TMMi,”. In: FIE. 2021.

[35] I. Costa and S. Oliveira, “A study on assets applied in design and execution activities of exploratory test to be used in teaching-learning: A survey application,” In: FIE. 2021.

[36] I. Costa and S. Oliveira, “A syllabus to Support Teaching and Learning of Exploratory Test Design and Execution,” 14th International Conference on Computer Supported Education - CSEDU. 2022.

[37] L. W. Anderson and D. R. Krathwohl, “A Taxonomy for Learning Teaching and Assesing: A Revision of Bloom's Taxonomy of Educational Objectives,” Longman. 2001

[38] A. P. Ferraz and R. V. Belhot, "Bloom's taxonomy: theoretical review and presentation of the instrument's adjustments to define instructional objectives," Gest. Prod. São Carlos, vol. 17, no. 2, pp. 421-431. 2010.

[39] Y. Chou, “Actionable Gamification - Beyond Points, Badges, and Leaderboards,” Octalysis Media. 2016.

[40] M. Pimentel and F. S. P. Carvalho, “Princípios da Educação Online: para sua aula não ficar massiva nem maçante!,” SBC Horizontes. ISSN 2175-9235. Available in: <http://horizontes.sbc.org.br/index.php/2020/05/23/principios-educacao-online>. 2020.

[41] F. Aguiar and P. Caroli, “Product Backlog Building (PBB): Concepção de um Product Backlog Efetivo,” Available in: <http://leanpub.com/pbb>. 2020.

[42] G. J. Myers, “The Art of Software Testing,” Third Edition. John Wiley & Sons, Inc.USA. 2011.

[43] J. Bach, “Exploratory Testing,” SATISFICE:. Software Testing For Serious People. Available in: <https://www.satisfice.com/exploratory-testing>. 2015.

[44] TestLink, “TestLink User Manual,” Available in: <http://testlink.sourceforge.net/docs/documents/end-users/manual.html>. 2022.

[45] J. Rooksby, J. Hunt, and X. Wang, “The Theory and Practice of Randori Coding Dojos,” In: 15th International Conference on Agile Software Development, pp 251- 259. 2014.

[46] L. C. F. Da Costa and S. R. B. Oliveira. “Um processo de Validação e Verificação em MPS com foco na implementação multi-modelo,” 17th International Conference on Information Systems and Technology Management - CONTECSI. ISSN 2448-1041. USP- São Paulo, Brasil.10.5748/17CONTECSI/PSE-6437. 2020.