Perception from the professors' point of view in the remote teaching of Software Testing using active methodologies during the Covid-19 pandemic.

Isaac Souza Elgrably
Graduate Program in Computer Science (PPGCC)
Federal University of Pará (UFPA)
Belém - PA, Brazil
isaacelgrably@gmail.com

Sandro Ronaldo Bezerra Oliveira
Graduate Program in Computer Science (PPGCC)
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/FIE56618.2022.9962481

***Abstract*—This Research Full Paper presents an analysis of the perception of research professors of Software Engineering on the use and difficulties in the context of developing classrooms with teaching strategies using active methodologies remotely. In recent years, due to the Covid-19 pandemic, professors have been facing several challenges to keep students engaged, motivated and learning. The active teaching methodologies that have been collaborating in recent years in the teaching and learning process of students, needed to adapt to the remote teaching class scenario that was implemented in several universities. This research measured the point of view of four professors, who are also researchers in computing education, about how their classes behaved in the last two years of the Software Testing subject in a graduate program at a Brazilian federal university. So that computing students have a way to obtain knowledge in a more practical way so that there is a decrease in the difficulty of students adapting to the job market. Professors are increasingly adopting active methodologies, group work and practical teaching approaches in order for students to acquire skills that will be decisive for their professional future. In view of the existing pandemic scenario and the need to adapt teaching plans for remote teaching, this paper sought to identify the perceptions that professors had, whether in the difficulties of adapting activities, adapting teaching plans to difficulties in keeping students attentive and motivated in the content that was being passed on. The objective of this work is to identify and discuss professors' perceptions about the use of active teaching methodologies using remote teaching in a software testing subject of a graduate class at a Brazilian federal university. It was also analyzed how the professors dealt with the issue of student motivation and engagement at that time. An analysis of the teaching plan of the software testing subject of the last two years of the graduate program in computing at a Brazilian federal university was carried out and each of the methodologies with the perception of the professors who participated in these subjects. The collection of these data was carried out by means of an interview with the four professors, the feedback from the students of the classes on each of the adopted methodologies was also analyzed and a triangulation of the data obtained was performed. The results indicate that there were great difficulties in adapting active methodologies from the point of view of the professors, however most of these challenges were overcome in the their views. It was also pointed out that certain methodologies and practices work better if the learning is done in an evolutionary way and that takes into account the students' previous knowledge about contents, tools and practices that the professors want to use. Thus, this paper aims to present an overview of professors' perspectives on the teaching of a computing subject that needs practical approaches using active methodologies in an emergency situation of remote teaching.**

Keywords— s***oftware testing, activities methodologies, remote teaching, educational innovation.***

I. Introduction

The situation that occurred due to the Covid-19 pandemic, in contrast to the need for the academic training of computer professionals continue to be able to the job market, became a determining factor for academic activities to continue even in the middle of the pandemic period. This reality required several adaptations to the teaching environment, especially for those subjects that require students to acquire more practical academic skills [1][2].

The MEC (Brazilian Ministry of Education) guides that in the subjects of computing courses, curricular contents can be taught in different forms of organization, according to pedagogical proposal, emphasizing teaching-learning methodologies, especially approaches that promote participation, collaboration and the involvement of students in the gradual constitution of their autonomy in the learning processes [3].

Software testing is a subject that is part of the Software Engineering knowledge area according to the SWEBOK (Software Engineering Body of Knowledge) guide [4]. Several current works present the importance that the teaching of software testing has been obtaining [2][5][6]. It is also highlighted that the software testing subject is an important activity of the software engineering process to ensure the development of high quality software [7].

Subjects with practical content, such as software testing, depend on the use of different teaching approaches so that the student can acquire and consolidate the knowledge that has been acquired, so the use of active teaching methodologies has helped in this process [2][7][8][9]. The need to teach these subjects remotely was a major challenge for computing professors.

So, this paper sought to investigate the teaching of software testing using active teaching methodologies from the point of view of professors to guide this objective, two Search Questions were built: (i) **Research Question 1 (RQ1)** - *Is it possible to guarantee the teaching and learning process of skills linked to software testing using active teaching methodologies remotely from the professors' point of view?* (ii) **Research Question 2 (RQ2)** - *Which teaching approaches are most harmed to teach in the form of online teaching taking into account the context of a practical subject, such as software testing?*

In order to solve these research questions, interviews were carried out and a structured feedback questionnaire was collected from four professors who taught software testing subjects in remote format using a syllabus [10] and a teaching plan [11] with active teaching methodologies such as foundation. The classes were taught in the years 2020-

2021 and 2021-2022 in a graduate program in computer science at a Brazilian federal public university. Another collaboration is the presentation of the way in which the professors constituted the teaching approaches used.

In addition to this initial section, this work was structured as follows: in Section II a theoretical foundation of the main themes of this work, all related to computing teaching, is provided, in Section III the research methodology that was carried out is described, presenting how it was once its organization, its operational flow and its objectives and results are done, Section IV presents the subject of software testing that served as a quasi-experiment for this research, mainly the planning of each of the teaching approaches used, in Section V the results collected from the professors on the functioning of the subject in a context of remote teaching are presented and discussed, in Section VI the conclusion of the work is presented, showing its main contributions, its limitations and possible threats to its validity and highlighting the future works that will come knowledge generated in this work.

II. Background

This section aims to provide a theoretical basis for the concepts used in this work.

A. Computer Teaching

There are several studies and guides that help to guide the teaching of computing subjects, some of them served as support for the construction of this work, as the report of the National Curricular Guidelines for computing courses in Brazilian territory points out that the teaching methodologies applied should stimulate the students to learn to solve problems, learn to learn, become independent and creative [3]. The student must have a central role in learning, he must be the protagonist and the professor the facilitator of this process.

In the SBC (Brazilian Computer Society) guide [12] it is exposed that a teaching methodology for students of computing institutes must be centered on the student as a subject of learning and not as a professor as a facilitator of the teaching-learning process. Extracurricular work should be used in such a way that the student learns to solve problems and is encouraged to learn, becoming independent and creative.

Furthermore, the CC (Computing Curricular) 2020 guide [13] exposes a serious problem in computing learning, in which technology now influences new ways of learning. Students use many non-traditional learning formats, thus challenging traditional methods. In addition, universities produce computer graduates who may be intellectually intelligent but struggle in work environments. Learning in computing education needs to incorporate knowledge along with other attributes.

The authors, using these reference guides as a basis, discuss in this work the difficulties of achieving what is listed for a good teaching-learning process in a remote teaching environment.

B. Active Teaching Methodologies

Active teaching methodologies are teaching strategies focused on the effective participation of students in the construction of the learning process, in a flexible, interconnected and hybrid way [14].

The teaching guides in the computing area propose that, whenever possible, active teaching methodologies should be used, so that the student spends more time in activities in which he is a protagonist in the teaching and learning process. In addition, professors must be prepared for the insertion of new paradigms that approach new curricular practices and teaching methodologies [11].

Active methodologies can be defined as a student-centered learning approach that allows them to be directly included in the entire teaching process, promoting criticality in problem solving and autonomy in the development of solutions [15]

Therefore, it is seen that active teaching methodologies have a broad alignment with important guides that guide the knowledge of computing subjects, and these guides are mentioned as the basis for the construction of pedagogical plans and academic curricula in Brazilian universities [16].

C. Remote Learning

Although research attributes greater ease for computing professors to adapt to remote teaching, in the same work it is pointed out that there is a great difficulty in promoting the learning process and ensuring students' interest in the subjects [17].

In [1] a sequence of good practices, findings and difficulties are indicated in a subject of software testing carried out remotely and it became clear that in the students' view some applied teaching methodologies ended up being harmed by the remote context.

In [18], which consists of a course project on model-based tests in remote teaching with a teaching context in software industry practices using flipped classrooms and then project-based learning, in this scenario there were also difficulties for the students to be motivated and to absorb the technical and practical knowledge that was being transmitted.

Thus, the authors see a need to analyze this same difficulty of remote teaching in the view of professors, from the conception of the teaching plan to the application in the classroom and verification of learning by students, especially when using active methodologies education.

III. Research Methodology

This work aims to present the results of an analysis from the point of view of professors of the use of teaching approaches with active teaching methodologies in a remote teaching context. In order to achieve the objective of the work, an interview was carried out with a structured questionnaire with four professors responsible for teaching this subject. The focal point of the questionnaire was to understand the point of view of these professors on each of the teaching approaches that were used.

This software testing subject is geared towards practical teaching. It uses an adapted version of a specific syllabus for teaching software testing [10] and the teaching plan [11]. It prioritizes the construction of knowledge based on academic competences and on content coming from different references, practical teaching activities and collaborative activities, in addition to presenting these contents with different active teaching methodologies, as suggested by the knowledge guides highlighted in Section II.

One of the points that made it possible to carry out this work was that, at each class, the participating professors met for a post-class, raising discussions and analyzing how each class was. A summary of these results can be seen in [1].

After collecting feedback from professors at each post-class at the end of each subject, questionnaires were passed on to the teaching professors to collect even more information on the challenges encountered in the subject as a whole. These results will be discussed in Section V.

IV. The Subject of Spftware Testing

The subject that is used in the quasi-experiment is a teaching subject of the Software Testing subtopic, which is part of the Software Engineering knowledge area following SWEBOK [4]. This subject is present in a graduate program in computer science at a Brazilian federal university and is optional, therefore, it is not mandatory for students.

This subject has a total of 32 classes and needed to be adapted for remote teaching due to the Covid-19 pandemic. This work presents the results of a total of two cycles of the subject, the first occurred in August 2020 to February 2021 with a total of 15 students and the other cycle in August 2021 to February 2022 with 13 students.

All participants in the quasi-experiment were duly enrolled in the course and in the first class they were informed of the course details. Altogether there were a total of twenty-eight students enrolled.

This Software Testing subject is based on the syllabus [10] which has four teaching units related to software testing:

- Software Engineering [SE]: it aims to deliver initial theoretical knowledge about concepts, models and practices about software testing, product quality and good construction practices,
- Software Development [SD]: its objective is knowledge related to the software construction and maintenance, identification and correction of bugs and good software development practices,
- Quality [QL]: it aims to provide knowledge about the quality of the software product presenting concepts and practices of testable projects, agile requirements, construction of test cases,
- Software Testing [ST]: its objective is to consolidate and put into practice the knowledge acquired in other teaching units. Here the tools and practices related to the topic are presented and students are challenged to implement them in practical projects.

The total content of the subject and each of the teaching units can be seen in [10]. This subject follows the indication of MEC [3] and uses competency-based learning as a line of evaluation of the teaching and learning process of students. Table I presents the academic competences and in which teaching units they should be learned.

However, the most important point of this work is the way in which a teaching plan was applied to teach this subject, then the teaching approaches will be presented with the active methodologies that were used in each of the teaching units of the subject.

A. *Teaching Approaches Used*

To propagate teaching and learning in competency-based learning, in addition to making the student the center of learning, the professors should adopt different teaching approaches that encourage different competences and generate different abilities in students so that there is a better chance of successful learning. Finally, the expected results and learning levels to be achieved for each teaching topic must be defined, where Bloom's revised taxonomy [19] was used in this work. Each of the teaching approaches will be detailed in subsections below.

TABLE I. Academic competences related to Teaching Units

Academic Competences	Teaching Units
Specify, design, implement, maintain and evaluate computing systems, employing appropriate theories, practices and tools.	Software Engineering, Quality, Software Testing.
Employ methodologies that aim to guarantee quality criteria throughout all stages of development of a computational solution.	Software Engineering, Quality, Software Testing.
Plan, specify, design, implement, test, verify and validate computer systems.	Software Development, Software Testing.
Understand and apply processes, techniques and procedures for the development, evolution and evaluation of software.	Software Engineering, Quality, Software Testing.
Evaluate the quality of Software Systems.	Software Testing.
Conceive, apply and validate principles, standards and best practices in software development.	Software Development, Quality, Software Testing.

1) Dialogued Expository Class: These are classes that have the structure of content presentation, containing: a historical summary and impact on the subject, presentation of the concept and ways of using the concepts, at least one example of the use of some of the concepts contained in the content of the class. This example may contain: presentation of tools and techniques by professors, graphics on the topic being presented. It is important that the professor has a role of facilitator and provokes the students so that there are discussions and personal and professional experiences on the topics covered.

As inputs, professors must expose the contents supported by practical examples, students must be participatory and exchange knowledge among themselves.

The dialogued classroom environment was a Google Meet virtual classroom and the support materials were present in a Google classroom. Students were encouraged to study part of the subject in advance, in order to be more participatory.

This teaching approach was used in all teaching units and was also used to achieve different expected results from Bloom's revised taxonomy, depending on the way the professors organized the class. Table II shows the expected results with the code of each teaching unit.

The evaluation of whether the expected results were achieved was made by observing the participation of students contributing to the exposition, asking, answering, following the understanding and analysis of concepts. In each of the classes, there were always at least two professors and one of them was responsible for this analysis and information collection.

TABLE II. EXPECTED RESULTS OF EXPOSITORY DIALOGUED CLASSES IN EACH TEACHING UNIT

	Factual	Conceptual	Procedural	Metacog-nitive
Remember	SE, SD, QL, ST			
Under-stand	SD, QL, ST	QL	ST	
Apply				
Analyze		ST		
Evaluate				
Create				

2) *Dojo Randori*: The Dojo Randori approach is a practice of teaching programming in a secure environment. The student will learn by trial and error, with the help of their peers.

In practice, a list of challenges about a real problem must be carried out, building its code and tests using the Test Driven Development (TDD) technique [20] and the Scratch online tool. The activity cycle is in which one student assumes the role of pilot and the rest of the students follow as an audience. He performs the coding for 5 minutes and the audience can assist him with comments. After five minutes, a student from the audience assumes the pilot position and the current pilot goes to the audience. The cycle repeats until all students have participated in the activity and then until the challenge is completed within the stipulated time of 2 hours. In the usual randori dojo cycle there was the role of the co-pilot who would assist the pilot and only he could talk to the pilot in the development, but this role had to be extinguished because of remote teaching, as the audience participants would be just spectators without much active participation.

The randori dojo environment was carried out in a Google Meet virtual classroom, in which when a student was a pilot he should share his screen and narrate its implementation, there was the addition of Scratch tool tutorials and implementation using TDD in the Google Classroom support materials. Students were encouraged to study part of the subject in advance, in order to be more participatory.

This teaching approach was used in the Software Engineering teaching unit in order to present practical and playful knowledge to students with concepts learned in the dialogued expository classes of the teaching unit. Table III shows the expected results with the code of each teaching unit.

TABLE III. EXPECTED RESULTS FROM DOJO RANDORI

	Factual	Conceptual	Procedural	Metacog-nitive
Remember				
Understand				
Apply				
Analyze		SE		
Evaluate	SE			
Create			SE	

The evaluation of whether the expected results were achieved was made from the observation of student participation, proactivity both as a pilot and as a member of the audience, and the final result delivered by all students was also corrected and part of the grade was given to all as a collaborative result.

3) *Dojo Kake:* The Dojo Kake approach is similar to a Dojo Randori activity, but instead of collaborating with the whole class, the whole activity is divided into multiple groups of 4 students, which working in parallel to carry out the activity. The professor has a more critical role within this context as they must solve the group's doubts over time. The groups have a list of challenges related to the implementation and maintenance of a software, all students in the group must alternate in the role of pilot and audience. This approach was used twice in the subject, in the first, the groups received the same initial code to address the challenges, aimed at finding and solving bugs, improving code quality and performing unit tests. In the second moment, the codes were exchanged between the groups and the new challenges would be carried out with the base code of another group.

The randori kake environment was carried out in multiple Google Meet virtual classrooms. The professors were distributed among the rooms and could provide mentoring, if necessary. Similar to the randori dojo, the student who was the pilot had to share his screen and narrate the his implementation. There was the addition of development tutorials in the Java language in the Google Classroom support materials. Students were encouraged to study part of the subject in advance, in order to be more participatory.

This teaching approach was used in the Software Development and Software Testing teaching unit in order to present practical and collaborative knowledge with open questions that would allow students to make different decisions on how to approach the challenge. Table IV shows the expected results with the code of each teaching unit.

TABLE IV. EXPECTED RESULTS FROM DOJO KAKE

	Factual	Conceptual	Procedural	Metacog-nitive
Remember				
Understand				
Apply		SE	ST	
Analyze	SE			
Evaluate		ST		
Create				SE, ST

The evaluation of whether the expected results were achieved was made from the observation of student participation, proactivity both as a pilot and as a member of the audience, commitment to link the activity to the previous knowledge that students already have and the final result was also corrected of delivery by the professors, checking if the objectives were achieved and giving a grade to each group.

4) *Flipped Classroom*: The flipped classroom approach is a hybrid teaching approach that proposes a logical inversion of knowledge internalization, in which students do a study on a certain topic of knowledge at home and then in the classroom together with the class, discusses the knowledge acquired and clears up possible content doubts with the help and guidance of the professor, who must play a role of facilitator and mediator of learning.

The approach was used only in the second quasi-experiment performed, after the first quasi-experiment class had mentioned in the feedback that it could add to the teaching and learning process. It was used twice during the course in the Quality and Software Testing units and focused on consolidating topics of knowledge that students had already seen before, but now the challenge was to bring some real use case of the content, which could be in personal or professional projects. Some topics of the contents of each of the units were separated and given to students who were divided into groups of up to 4 students.

The class environment was in a Google Meet virtual classroom. Students searched for knowledge in the references of support material, external projects, works and elsewhere. After each group had 1 hour to present and discuss with the rest of the class what they had collected and presented. Each student was encouraged to study the topic of the other groups to improve participation.

Table V shows the expected results with the code of each teaching unit.

TABLE V. EXPECTED RESULTS OF THE FLIPPED CLASSROOM

	Factual	Conceptual	Procedural	Metacognitive
Remember				
Understand		QL, ST		
Apply				
Analyze		QL	ST	
Evaluate		ST	QL	
Create				

The evaluation of whether the expected results were achieved was made from the observation of students' participation in the presentations, answering questions from other students in their presentation, bringing questions and enriching the content presented by another group and the content of the presentation delivered.

5) Project-Based Learning: In carrying out practical activities, the main objective is to promote learning focused on delivering experience that simulates the real world for participants, making them acquire skills that are important in the existing challenges in the job market.

This teaching approach was divided into two moments. Initially a web system was delivered to each of the groups, in the first moment in the Quality unit the groups should carry out tests to collect errors, failures and possible new requirements and in the second moment in the Software Testing unit should be built a prototype of system corrections and a document of test cases for the problems found in the first moment.

The students had around 4 weeks to carry out each of the moments and at the end of the period they had to present their findings to the whole class in a 30-minute presentation per group. This is a practical activity that simulates a real test environment, so it has many expected results.

Table VI shows the expected results with the code of each teaching unit.

The evaluation of whether the expected results were achieved was made based on the observation of the presentation of what was built and collected and the correction of all developed work products.

TABLE VI. EXPECTED RESULTS OF PROJECT-BASED LEARNING

	Factual	Conceptual	Procedural	Metacog-nitive
Remember				
Under-stand				
Apply			ST	
Analyze		ST	QL	
Evaluate		QL		
Create	QL			ST

For this work, there was no need to fully detail the functioning of each class and each activity, nor to detail the content that was taught. More details on the functioning of one of the quasi-experiment groups can be seen in [2].

V. DISCUSSION AND RESULTS

After presenting the modeling and objectives of each of the teaching approaches used, we discuss the results of this work.

It is important to point out that these professors who answered the questionnaire taught the subject, but are not authors of the work. A total of 25 questions were asked to the professors, there were 5 basic questions for each of the different teaching approaches, all with a Likert 5 scale. For all Likert 5 scale items, the coefficient of internal consistency of Cronbach's Alpha [21] was considered "Excellent" with α = 0. 0.943407777 ($\cong$ 0.94), this makes it possible for the results to be widely compared and correlated. Table VII presents the 5 basic questions that were replicated in the five teaching approaches used, thus totaling the 25 questions asked to professors.

TABLE VII. PROFESSORS' PERCEPTION OF TEACHING APPROACHES

Item	Question
I1	Construction of a subject with the approach
I2	Evaluate student participation in the approach
I3	Make students interact during the approach
I4	Ease of collecting student feedback
I5	The approach manages to achieve the learning objectives in a concrete way

Each of the questionnaire items is on the Likert Scale from 1 to 5, where the scale assignments were: 1 - Very Easy, 2 - Easy, 3 - Medium, 4 - Difficult and 5 - Very Difficult. Each of these items was evaluated for each of the teaching approaches: dialogued expository classroom, dojo randori, dojo kake, flipped classroom and project-based learning. For better understanding the discussion will be divided into a subsection for each teaching approach.

A. Dialogued Expositive Class

About the dialogued expository class, the 5 items present in Table VII were evaluated by all professors, following the Likert scale described after Table VII of this section. The results can be seen in Figure 1.

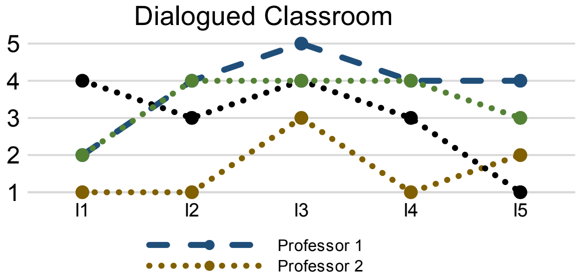

Fig. 1. How do you evaluate the following items regarding the dialogued expository class in a remote teaching context?

In addition to answering the degree of difficulty of each item, professors had spaces to comment on strengths, weaknesses and improvements to be made in each approach.

Professors scored between 1 - Very Easy and 4 - Difficult on the topic of building the teaching approach in a remote environment. The biggest problem highlighted by the professors is in maintaining student engagement. It was noticed that at some points in the course only a few students were engaged in answering and raising questions about the points raised by the professors. Professors were unanimous in stating that a Software Testing subject, for having very specific topics, needs to have this type of teaching approach.

Regarding the evaluation method during the approach, the professors' answer was between 1 - Very Easy and 4 – Difficult. One of the professors commented that he considers remote teaching better to collect student feedback, due to the possibility of recording the class and chat. Another professor found it difficult to evaluate the full participation of students and difficult to manage the debates.

Regarding the possibility of making students interact during the approach, the professors considered from 3 - Medium to 5 - Very Difficult and justified it by saying that there are many noises and distractions during a remote class, mainly due to the limitation of the local internet that makes it necessary for students turn off the cameras.

Regarding the ease of collecting feedback from students, most professors found it 2 – Easy or 3 – Regular and it was commented that Google Classroom allows feedback and perception questionnaires to be passed constantly and gamification can be used to encourage this participation of students.

Finally, regarding the difficulty of the approach to achieve the learning objectives, the vast majority of professors highlighted that it would be 2 - Easy because it is supported by dynamics that encourage the practical involvement of students with the topic studied and maintain an organization of the content that stimulates both visual and auditory perception. Students tend to consume support materials and seek knowledge before going to the dialogued expository class.

B. *Dojo Randori*

About the randori dojo, the 5 items present in Table VII were also evaluated by all professors, following the likert scale explained at the beginning of the section. The results can be seen in Figure 2.

The professors had discrepant evaluations about the construction of a subject with the approach and gave answers between 2 - Easy and 5 - Very Difficult. They mentioned that depending on the subject they do not see the condition to pass the practice remotely and that with a greater number of that 10 students make it very difficult to organize the practice.

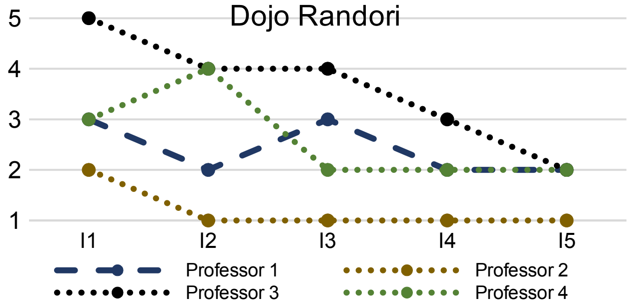

Fig. 2. How do you evaluate the following items regarding Dojo Randori in a remote teaching context?

Professors had discrepant evaluations on the ease of evaluating student participation in practice remotely. They mentioned that many students end up participating only once and then leaving the practice to the more engaged students.

Regarding the ease of student engagement in practice, professors' evaluations were between 1 - Very Easy and 4 Difficult and it was commented that great moderation and mediation of professors is necessary and especially that one or more students take the lead in the practice.

Regarding the possibility of collecting feedback, the answers were mostly positive since the practice is recorded and the students have autonomy to participate and the solution is collective and it is noticeable even if any student is not actively participating in the discussion.

About the difficulty of the approach to achieve the learning objectives, professors were unanimous that it was easy to perceive: three professors answered 3 – Easy and one professor answered 1 – Easy. The requirement that students have to participate at least once as a pilot already gives a possibility of individual evaluation and during the practice the professor can provoke the audience to answer more questions and question the progress of the practice, where the knowledge of the students becomes visible about the content involved.

C. *Dojo Kake*

About the dojo kake, the 5 items present in Table VII were also evaluated by all professors, following the likert scale explained at the beginning of the section. The results can be seen in Figure 3.

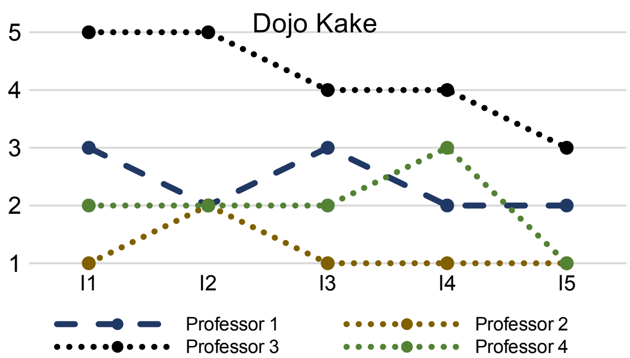

Fig. 3. How do you evaluate the following items regarding Dojo Kake in a remote teaching context?

The professors were divergent in the answers about building a classroom with dojo kake, but most did not consider it difficult in the remote context. They commented that it requires a high degree of planning, some subjects would need specific tools and it is difficult to provide mentoring for students during practice.

About the ease of evaluating students, only one of the professors disagreed that it would be easy. He argued that better control of learning indicators is necessary, because from a certain moment the best programmer in the group ends up staying a long time as a pilot.

For the collection of feedback, the professors answered differently, but only one considered it 4 – Difficult. The professors consider that the recording of the practice and when a version control is carried out via Github of the delivered code, makes it possible to see the participation of the students in the conclusion of the task.

The last analysis of the dojo kake is the difficulty of achieving the learning objectives. In this question none of the professors considered it difficult. Considering that the rooms are live on the meet during practice and the pilot always shares the screen and narrates his task, it is considerably easy to see if the students achieved the expected results that were listed. One of the professors highlighted that it is even more easier than just reading the code delivered by the group, as the professors record the video of each room and can check it later.

D. Flipped Classroom

Regarding the flipped classroom, the 5 items present in Table VII were also evaluated by all professors, following the Likert scale explained at the beginning of the section. The results can be seen in Figure 4.

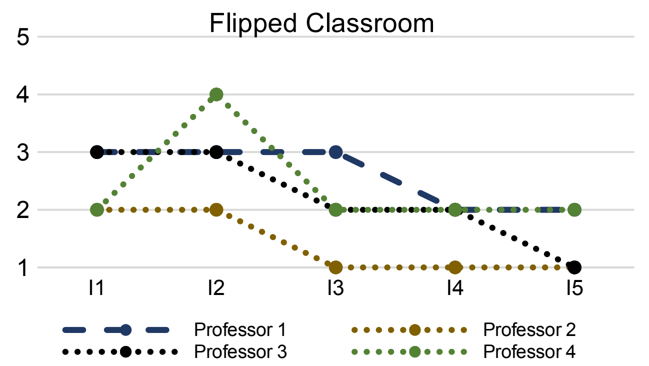

Fig. 4. 6 - How do you evaluate the following items regarding the Flipped Classroom in a remote teaching context?

Professors consider a fully remote flipped classroom to be 2 - Easy or 3 - Medium difficulty in construction. They said that they found a certain low degree of engagement in the activity and it focused only on a few members of each group.

As for the possibility of evaluating participation, the practice was not so simple, as the groups really centralized the tasks in some members and the professor had to provoke the other members of the group for them to participate and help in the group's practice. This is a point that the authors found it interesting that the flipped classroom is a hybrid approach to teaching and should be easily adapted to remote teaching, but the points raised by the professors were real and difficult to solve at the time.

Regarding the ease of making students interact during practice, none of the professors found it difficult, but they commented that they expected it to be easier than it actually was, as the groups seemed to avoid debating among themselves, especially when the group that was presenting did not properly delved into its topic.

The collection of feedback in this approach was considered extremely easy by the professors, because as the groups had to bring the knowledge to the classroom. It was very visible when the work was well done and the doubts and questions that were raised.

The difficulty of achieving the learning objectives listed was considered easy or very easy by the professors and they highlighted that it was visible when the students mastered the topic they were presenting and the duration of 1 hour for each group made it possible for the professors to question the group they were presenting and new guidelines are raised on the subject, to analyze the depth of knowledge that has been achieved.

E. Project-Based Learning

Regarding project-based learning, the 5 items in Table VII were also evaluated by all profesors, following the Likert scale explained at the beginning of the section. The results can be seen in Figure 5.

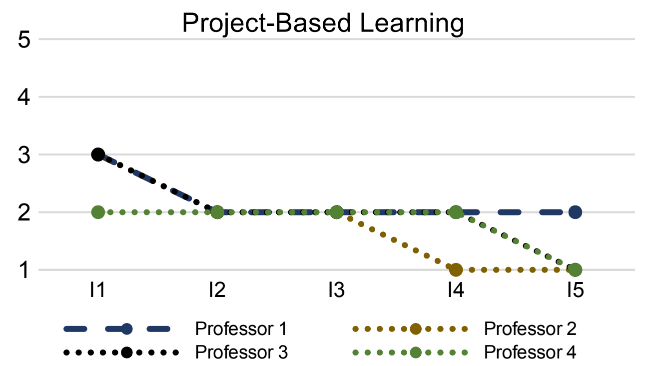

Fig. 5. How do you evaluate the following items regarding the Project-Based Learning in a remote teaching context?

Professors considered the difficulty of building a learning practice based on projects between 2 – Easy and 3 – Medium in the remote context and commented that for practical computing subjects such as software testing, the remote teaching period ended up benefiting some students to acquire knowledge about Devop's practices that ended up facilitating the practice of a real problem being delivered.

As for the possibility of evaluating participation, all professors considered 2 - Easy, as the engagement and participation of the students was very high, thus facilitating the professors' tasks. The fact that it was done asynchronously also made it possible for professors to mentor students which had a very positive weight on participation.

Regarding the ease of making students interact, all the professors found it easy to verify and stimulate, as the approach ended up being very well evaluated by the students as well. The professors described that one of the possible factors was that the activity was the highest score in the subjects and another that the activity simulating a real environment was a factor of engagement very well used by the students.

Collecting feedback in this approach was considered extremely easy by professors as students sought contact with professors, scheduled mentorships and gave weekly feedback on their progress. Some groups set up Kanban boards to mark their progress and added professors as participants.

The difficulty of achieving the learning objectives was considered quite easy by the professors. Even if the objectives were considered open questions the great commitment of the groups and the demonstration of their skills and competences acquired previously in the subject and before it were factors that made the evaluation of the work products very easy, even though it is done judiciously by the professors.

VI. Conclusion

Initially, we see that the two research questions of this work "RQ1: *Is it possible to guarantee the process of teaching and learning skills linked to software testing using active teaching methodologies remotely from the point of view of professors?*" and "RQ2: *What are the most impaired teaching approaches to teach in the form of online teaching taking into account the context of a practical subject, such as software testing?*" can be translated through feedback from the professors who taught the course.

Regarding RQ1, mainly in line with I5 (in Table VII) present in all the professors' questionnaires to evaluate each teaching approach, only the dialogued expository classes were considered by a professor as difficult to verify that the students had achieved the skills related to software testing to this quasi-experiment in a remote teaching environment.

However, we must also point out that professors considered that organizing approaches and maintaining student engagement something difficult in some practices with active methodologies. This is due to the fact that when an active methodology focuses on practical and collaborative learning, the distance ends making the professor's perception of the exchange of knowledge between students a little difficult and as the delivery of the practice is in a group there may be the possibility that some of the students have not learned the contents properly.

About RQ 2, in the view of the professors, dojos and the dialogued classroom are the approaches most affected in a remote context. The dojos lose a lot of the quality of the pairing, they lose the important role of the co-pilot and the audience doesn't talk much among themselves, but only with the pilot at times, this ends up reducing the pilot's creative freedom in practice. The dialogued classroom is also quite harmed, especially in terms of engagement and discussions, many students end up abstaining from participating and the possibility for the professor to provoke the student in this remote format is limited, given that sometimes the students are only online and are not participating there. Gamification with punitive elements is one of the approaches that can help maintain student participation.

terrible idea

A. Contribuitions

The main contribution of this work is the presentation of the point of view of different professors who have to teach a subject that uses active teaching methodologies in a remote teaching context. Several works [1][2][17][18] present the students' point of view in the remote teaching and learning process, so this work sought to add the professors' point of view to the choice of analyzing approach by approach aimed at of being able to separate the learning context and reduce the scope of the results, since the results of the evaluation of each teaching approach are quite different. This generated knowledge can be used by other professors, course coordinators or researchers who want to build practical subjects supported by active teaching methodologies.

B. Limitations

There are some limitations in this work, the results obtained must be treated in a controlled way, it will be difficult to generalize the results widely. Each of the professors who participated in this work has a master's degree in computer science and a researcher in Software Engineering and teaching and education in computing and all participated in a subjects with the scope presented. Another factor is that the subjects is taught in a postgraduate course in computer science and the participating students already have considerable professional and practical experience in Computing and Software Engineering. This may have favored participation and engagement in certain practices, which was reflected in the professors' feedbacks.

C. Future Works

As future works, the authors intend to insert this questionnaire of professors' views on teaching approaches in the next experiments, quasi-experiments and case studies of the research group that they are part of.

Feedback will also be collected from professors who are teaching active teaching methodologies in person and also did so remotely, so that we can advance in the difficulties already highlighted in [1] and [17], in addition to also verifying some points that the professors commented that it was easier to evaluate remotely, information that surprised the authors.

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] I. S. Elgrably and S. R. B. Oliveira, "Remote teaching and learning of software testing using active methodologies in the COVID-19 pandemic context," 2021 IEEE Frontiers in Education Conference (FIE), 2021, pp. 1-9, doi: 10.1109/FIE49875.2021.9637426.

[2] I. S. Elgrably and S. R. B. Oliveira, "A Quasi-Experimental Evaluation of Teaching Software Testing in Software Quality Assurance Subject during a Post-Graduate Computer Science Course", Int. J. Emerg. Technol. Learn., vol. 17, no. 05, pp. pp. 57–86, Mar. 2022.

[3] MEC. Diretrizes Curriculares Nacionais para os Cursos De Graduação em Computação (Dcn16). 2016.

[4] P. Bourque and R. E. Fairley.SWEBOK Guide V3.0. 2014. Available: www.swebok.org.

[5] V. Garousi, A. Rainer, P. Lauv°as Jr, and A. Arcuri, "Software-testing education: A systematic literature mapping," J. Syst. Softw., vol. 165, p. 110570, 2020.

[6] T. Rahman, J. Nwokeji, R. Matovu, S. Frezza, H. Sugnanam and A. Pisolkar, "Analyzing Competences in Software Testing: Combining Thematic Analysis with Natural Language Processing (NLP)," 2021 IEEE Frontiers in Education Conference (FIE), 2021, pp. 1-9, doi: 10.1109/FIE49875.2021.9637220.

[7] B. Gopal and S. Cooper, "Peer Instruction in Software Testing and Continuous Integration," Proceedings of the 52nd ACM Technical Symposium on Computer Science Education. ACM, Mar. 03, 2021 [Online]. Available: http://dx.doi.org/10.1145/3408877.3432404.

[8] I. E. Ferreira Costa and S. R. Bezerra Oliveira, "A mapping of prescribed assets in the ACM / IEEE and SBC curriculum to the test design and execution practices of TMMi," 2021 IEEE Frontiers in Education Conference (FIE), 2021, pp. 1-8, doi: 10.1109/FIE49875.2021.9637410.

[9] L. P. Scatalon, R. E. Garcia, and E. F. Barbosa, "Teaching Practices of Software Testing in Programming Education". IEEE Frontiers in Education Conference, Sweden, 2020.

[10] I. S. Elgrably and S. R. B. Oliveira, "Construction of a syllabus adhering to the teaching of software testing using agile practices". IEEE Frontiers in Education Conference, Sweden, 2020.

[11] I. S. Elgrably and S. R. B. Oliveira, "Model for teaching and training software testing in an agile context," 2020 IEEE Frontiers in Education Conference (FIE), 2020, pp. 1-9, doi: 10.1109/ FIE44824.2020.9274117.

[12] F.A. Zorzo, D. Nunes, E. Matos, I. Steinmacher, J. Leite, R. M. Araujo, R. Correia, and S. Martins, "Training References for Undergraduate Computing Courses". Sociedade Brasileira de Computação (SBC), 2017.

[13] CC2020 Task Force, "Computing Curricula 2020." ACM, Nov. 15, 2020. doi: 10.1145/3467967.

[14] J. Moran, ''Active methodologies and hybrid models in education,'' in New Digital Technologies: Reflections on Mediation, Learning and Development, S. Yaegashi et al., Eds. Curitiba, Brazil: CRV, 2017, pp. 23–35

[15] M. Hernández-de-Menéndez, A. Vallejo Guevara, J. C. Tudón Martínez, D. Hernández Alcántara, and R. Morales-Menendez, "Active learning in engineering education. A review of fundamentals, best practices and experiences," International Journal on Interactive Design and Manufacturing (IJIDeM), vol. 13, no. 3. Springer Science and Business Media LLC, pp. 909–922, Feb. 13, 2019 [Online]. Available: http://dx.doi.org/10.1007/s12008-019-00557-8.

[16] I. S. Elgrably and S. R. B. de Oliveira, "A diagnosis on software testing education in the Brazilian Universities," 2021 IEEE Frontiers in Education Conference (FIE), 2021, pp. 1-8, doi: 10.1109/ FIE49875.2021.9637305.

[17] Sarmento, C. F., Rique, T. P., Nascimento, S. M., & Lira, R. V. (2021). Investigando os impactos da COVID-19 no ensino remoto da computação: uma análise no nordeste do Brasil / Investigating COVID-19's impacts on remote computing teaching: an analysis in northeastern Brazil. In Brazilian Journal of Development (Vol. 7, Issue 6, pp. 64072–64099). South Florida Publishing LLC. https://doi.org/10.34117/bjdv7n6-676.

[18] E. P. Enoiu, "Teaching Software Testing to Industrial Practitioners Using Distance and Web-Based Learning," Frontiers in Software Engineering Education. Springer International Publishing, pp. 73–87, 2020 [Online]. Available: http://dx.doi.org/10.1007/978-3-030-57663-9_6.

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

[20] K. Beck. TDD - Desenvolvimento Guiado por Testes. 1. Ed. Porto Alegre, Bookman Editora, 2010.

[21] L. Cronbach, "Coefficient alpha and the internal structure of tests". Psychometrika, n. 16, 1951.