Using flipped classroom to promote active learning and engagement in a Software Testing subject remotely 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/FIE56618.2022.9962379

***Abstract*—This Research to Practice Full Paper presents the results obtained in an implementation of a flipped classroom, taught remotely synchronously in a subject of software testing in a graduate class. Due to the maintenance of the COVID-19 pandemic in the year 2021 due to the emergence of some variants, some universities around the world, especially in countries with a still high transmission rate, decided to continue with remote classes in this academic period. In this context, Software Testing is an important topic of knowledge derived from Software Engineering and has been gaining prominence in development companies in recent years, however it is a subject that needs a more practical and dynamic way of learning to obtain skills and competences. by the students. In order to promote a good teaching and learning process, more and more teachers and professors have been taking greater care in the selection of teaching strategies to help build the knowledge of their students, one of these strategies is the flipped classroom, in which the student internalizes the contents pointed out before a class and then, together with the class, a discussion is held about the knowledge acquired with the help and guidance of the teacher / professor. However, this knowledge transfer process has some extra difficulties with content that are more easily taught in a practical and face-to-face manner. Assuming, this work presents how the authors got around these difficulties and what the students present in a graduate class in a Software Testing subject thought of the adopted teaching strategy. To present an experience report of the use of the flipped classroom strategy in a software testing subject in a graduate computer science class. In parallel also, analyze the motivation and engagement of students in this learning scenario of the strategy in remote teaching. For this subject, a specific curriculum was selected for the teaching of software testing that takes into account the academic competences of the Brazilian Ministry of Education – MEC. Both the learning of the content were evaluated, taking into account the evaluations that were used in the flipped classroom with other teaching strategies in two other previous classes, the feedback given by the students about the subject and specifically about this teaching strategy was also considered, which was obtained via a structured questionnaire answered anonymously by each of the students participating in the course. The results indicate that there was an increase in the students' grade compared to evaluations of the same subject in the two previous years. However, the most important result was the feedback given by the students in the perception questionnaire of each teaching strategy adopted in the subject, considering that there was an excellent acceptance according to the participants. This paper mainly aims to help teachers / professors and researchers of active teaching methodologies in the perspective of knowledge construction in remote teaching using the flipped classroom as a support strategy.**

Keywords—flipped classroom, software testing, remote teaching

I. INTRODUCTION

Brazilian federal universities had to deal with a year 2021 restricted to applying the teaching and learning process remotely, due to the continuity of the COVID-19 pandemic. This teaching strictly done remotely in addition to the reclusion and necessary distance becomes a complicated challenge for the active learning process and the engagement of university students, especially in those subjects that require more practical and collaborative ways of learning, such as the subject of software testing [1] [2] [3] [4] .

In the literature, several authors point out that the subject of software testing needs to be taught in a more active and practical way so that students acquire skills and can stimulate necessary skills in the industry [5]. In addition, some authors emphasize that the stimulation of practical skills (soft skills) and experience in collaborative activities are crucial for students who want to be testing professionals [6] [7].

The CC2020 guide [8], in its appendix, presents a combination of learning and practice to develop skills for higher education students and among these teaching strategies, the flipped classroom has been one of the teaching strategies that have been used for the teaching computing topics by university professors, as seen in Fioravanti and Barbosa [9], it proposes an inversion of the environments in which learning activities are carried out. The explanation and search for content takes place at home, using video lessons and other resources provided by the teacher, and the resolution of exercises and other activities now take place in the classroom.

The objective of this paper is to show student engagement using a flipped classroom cycle proposal for knowledge construction in a software testing subject in a computer science graduate class at a Brazilian federal public university. The subject was modeled taking into account the academic content present in a specialized curriculum for teaching software testing [10].

To evaluate student engagement on the flipped classroom and its role in building the student learning process, the following research question (RQ) was created: *Is it possible to ensure engagement in a software testing course using flipped classrooms in a remote learning scenario?*

For the resolution of the mentioned research question, a feedback questionnaire was evaluated in a quantitative and qualitative way from a quasi-experiment carried out with graduate students of a subject of software testing. The results obtained will be compared to the other methodologies used. Another contribution of this paper is the presentation of the way the authors approached the teaching strategy of the flipped classroom, taking into account several factors, such

as: remote teaching, active teaching methodologies, collaborative learning, among others.

In addition to this introductory section, this paper is structured as follows: Section II presents the theoretical foundation that supports the development of this research, Section III presents the context of the subject used as a basis for the development of the research, Section IV presents the results of the quasi- experiment carried out in the research and Section V presents the conclusions of the research carried out.

II. Background

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

A. Software Testing

Software testing is a fundamental activity to ensure a certain degree of quality in software systems [11]. Software testing is quite necessary as it provides a validation method for developers, demanding to ensure that the software meets certain requirements or standards, which can be used by any organization [12].

If a software does not work properly it can lead to many problems, including loss of money, time or business reputation, and even injury or death. Software testing is a way to evaluate software quality and reduce the risk of software failure in operation [13]. When we consider critical security software, costs and efforts increase. Therefore, it is essential to have a good testing strategy for any software development industry [14].

According to the TMMi (Test Maturity Model Integration) model, the importance of the test subjetc, as one of the quality measures that can be taken, is growing rapidly. Testing has become a key activity that directly influences not only product quality, but also the "performance" of the entire development and manufacturing process [15].

It is verified that software testing is a fundamental activity for the construction of a good software product and it has become important that students have the knowledge of its consolidated contents.

B. Teaching Approaches and Strategies

The CC2020 guide [8] mentions that universities should explore new learning methods that can increase knowledge learning and allow students to interact with each other to develop new skill sets, as well as develop communication and teamwork skills by studying. with other people.

The SBC 2017 guide [16] suggests that teaching approaches for computing students in Brazilian institutions should be centered on the student as a subject of learning and supported by the teacher 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 to learn, becoming independent and creative.

Also in the SBC 2017 guide [16] it is said that the teacher must present the applications of theoretical content, be a mediator, stimulate competition, communication, provoke teamwork, motivate students to study and guide reasoning and develop communication and negotiation skills. This suggestion framework is closely aligned with the flipped classroom teaching strategy.

C. Flipped Classroom

Flipped classroom means flipping the classroom, which proposes that events that traditionally take place inside the classroom now happen outside the classroom and vice versa [17]

According to Qu, Cai and Haj-Hussein [18], for students of computing or information technology this learning format fits very well. The flipped classroom format allows students enough time to familiarize themselves with the class materials on their own and then fully utilize the time where they can communicate directly with instructors about questions they may have.

The flipped classroom model integrates active, collaborative learning practices into synchronous class time and allows students to gain exposure to new concepts individually and asynchronously, often in the form of pre-class video lectures and low-level risk evaluations [19]. For Quaresma and Oliveira [20], in the flipped classroom teaching strategy, the teacher has the role of knowledge mediator, while the student has the role of knowledge propagator.

Thus, the authors of this work chose to add this teaching practice to the second version of a set of a sequence of quasi-experiments on teaching tests with active teaching methodologies [21] [22], seeing the opportunity to add and adapt the practice in a fully remote teaching environment, due to the COVID-19 pandemic.

III. The Subject of Software Testing

The subject that is used in the quasi-experiment presented in this paper happens annually in a graduate computer science program and is optional. It can be carried out by master's, doctoral or special students, these are those students who seek to contemplate some subjects before joining the postgraduate program. The course lasts for 1 semester with a total workload of 60 hours, with 32 classes during the academic semester, each lasting up to 2 hours. In 2021, the class had a total of 13 students enrolled.

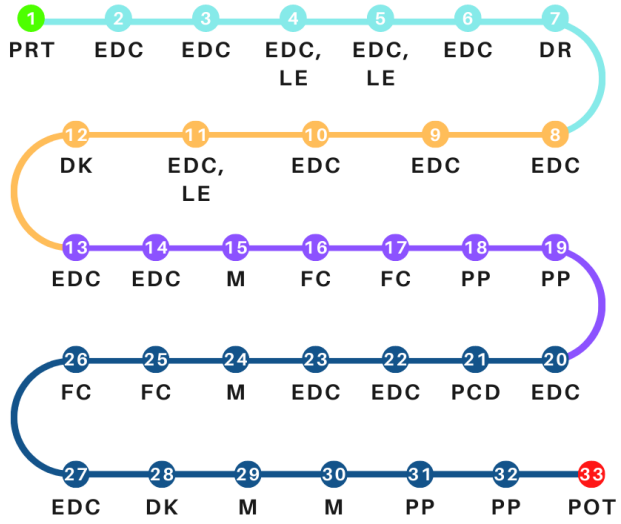
A. Subject Configuration

The subject was designed to emphasize learning focused on active teaching methodologies [24] on the contents related to software testing, which can be seen in Table I.

TABLE I. Adapted from Syllabus Of Software Testing [10]

Programmatic Content (PC)
Teaching Unit 1 - Software Engineering
1.1 Introduction to testing and quality
1.2 Introduction to software development and its development methods
Teaching Unit 2 - Software Construction
2.1 Software development concepts
2.2 Defining and implementing defensive programming and software maintainability techniques
Teaching Unit 3 - Quality
3.1 Software quality concepts
3.2 Software quality strategies aligned to testing
Teaching Unit 4 - Software Testing
4.1 Initial concepts for developing testing approaches
4.2 Construction of work products for validation and verification of software testing
4.3 Test-oriented development project
4.4 Evaluation of test-oriented project

Some of the course's learning objectives are related to understanding software testing techniques and practices and product quality, building software testing work products, developing code following good coding practices, and evaluating and maintaining computer systems. All the learning objectives can be seen in [2]. Learning is evaluated via Bloom's revised taxonomy [22] verifying that students have achieved the expected learning outcomes. In Figure 1 the script of the subject is presented.



Legend:

PRT	Pré-Test	LE	List of Exercicies
	Software Engineering	DR	Dojo Randori
	Software Construction	DK	Dojo Kake
	Quality	M	Mentoring
	Software Testing	FC	Flipped Classroom
POT	Pós-Test	PP	Practical Project
EDC	Expositive Dialogue Class	PCD	Practical Case Discussion

Fig. 1. Software Testing Subject Roadmap.

This paper will focus on the aspect of using the flipped classroom that was addressed in the Quality and Software Testing units of the subject, always being preceded by a mentoring class with the teams, we will also see the correlation with the other methodologies in the construction process knowledge of participating students.

B. *Flipped Classroom Configuration*

The flipped classroom approach to the subject happened in two different moments, one of them in unit 3 of Quality and another in unit 4 of Software Testing. The decision of the moment to apply the approach was due to the fact that the professors prefer that the students have already consolidated an initial theoretical knowledge about the contents via dialogic expository classes.

The task for the flipped classroom had different peculiarities in each of the moments, in the first moment in the Teaching Unit 3 – Quality, the teaching approach worked as follows:

- Students were divided into four teams,
- Four themes were drawn for the teams: I = Software Architecture, II – Software Project, III = Design Standards and IV = Requirements Design,
- Each of the teams' themes should be correlated with some software testing content,
- Each team would have half a class (1 hour) to present its content in which at least 10 minutes should be for discussion with the other teams and teachers,
- As a prerequisite, each team, in addition to the content, should present some tool and at least one practical example of the content presented,
- Teachers played the role of mentors: indicating references, books and video lessons, in addition to meeting with the teams individually to solve doubts, guide on the use of practices and tools and evaluate the compliance of the contents.

In the second moment of the flipped classroom in Teaching Unit IV – Software Testing, the teaching approach worked as follows:

- Students remained in the same four teams,
- Four themes were drawn for the teams: I = Non-Functional Tests, II – Estimation of Time and Risk in Software Tests, III = Test Coverage and IV = Process Improvement in Tests,
- The themes of this second flipped classroom are software testing contents that are important for the software industry [4] and that Brazilian institutions rarely address, as can be seen in [1],
- Again, each team would have half a class (1 hour) to present their content in which at least 10 minutes should be for discussion with the other teams and teachers,
- Unlike the first flipped classroom, teams would not need to present tools, but should bring some case study of their work or personal project in which a team member has already participated, related to the topic.
- The teachers played the role of mentors: indicating references, books and video lessons, in addition to meeting with the teams individually to resolve doubts, advise on the projects developed and evaluate the compliance of the contents.

In Figure 2 it can see a flow of the functioning of the flipped classroom in the subject. In prior classes to this one, the contents of these classes had already been disseminated, via dialogic expository classes, the use of the flipped classroom aimed to develop skills, such as: autonomy, problem-solving ability and critical thinking to students.

The expected results for the first flipped classroom were that students were able to understand the contents previously taught, analyze problems related to the contents and evaluate situations of applicability of practices and tools in software testing. As for the second flipped classroom, the expected results were that students must understand the functioning of tests in the life cycle of a software, must be able to analyze work products, realizing if their practices are functional and evolutionary and can evaluate software solutions or products criticizing their construction and the techniques, practices and approaches chosen.

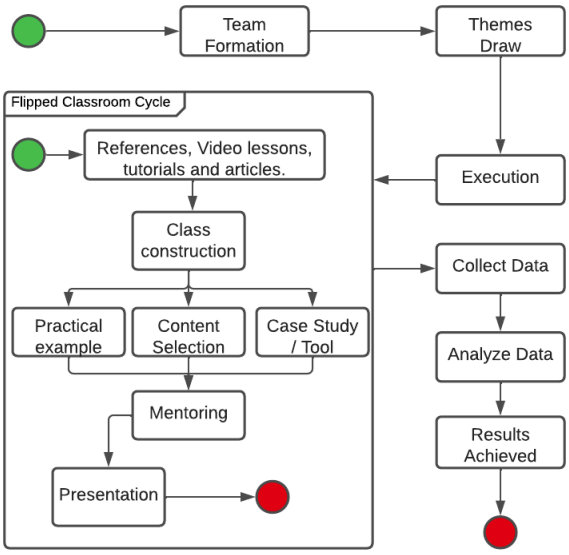


Fig. 2. Workflow of the Flipped Classroom.

According to bloom's revised taxonomy [22] the expected cognitive and knowledge levels that students reach are those that can be seen in Table II. Flipped classroom 1 has the legend of FC1 and the flipped classroom 2 has the legend of FC2.

TABLE II. EXPECTED LEARNING OUTCOMES

	Factual	Conceptual	Procedural	Metacog-nitive
Remember				
Understand		FC1 / FC2		
Apply				
Analyze		FC1	FC2	
Evaluate		FC2	FC1	
Create				

It is important to clarify that the same teaching approach may have different expected results in Bloom's revised taxonomy [22]. This is possible through different commands that were given to students in the practices, so the configuration of each one of the flipped classrooms that were held was shown.

The engagement results of the students' perception of the flipped classroom approach were collected at the end of the subject in a questionnaire with quantitative and qualitative questions and sent blindly by the students, in which the students were informed that they would not know who would be the author of the answer to the questionnaire later these results were analyzed and are being presented and discussed in this work. Finally, in the next sections the results will be presented and discussed by the authors.

IV. RESULTS IN QUASI-EXPERIMENT

The RQ of this paper is about engagement in learning using the flipped classroom approach for a graduate class in a subject of software testing in a remote teaching environment. In this context, the authors evaluated aspects of the flipped classroom approach that was built for this quasi-experiment. A total of 5 questions, all with a Likert 5 scale. For all Likert 5 scale items, the Cronbach's Alpha internal consistency coefficient [23] was considered "Acceptable" with $\alpha = 0.750786164$ ($\cong 0.75$) that allows us to analyze the results in parallel with a certain degree of confidence in the results. In Table III the items present in the questionnaire are presented, for each of these items the students should make a comment in text justifying their answer.

TABLE III. QUESTIONS INCLUDED IN THE FLIPPED CLASSROOM QUESTIONNAIRE

Item	Question
1	How do you evaluate the flipped classroom in relation to the scope of the activity?
2	How do you evaluate the flipped classroom in terms of the difficulty of the activity?
3	How do you evaluate the flipped classroom in terms of activity duration?
4	How do you evaluate the flipped classroom in relation to the knowledge of the subject having served as an aid?
5	How do you evaluate Flipped Classrooms in relation to learning subject content?

For the first five items of the questionnaire, questions were asked to be answered on the Likert Scale from 1 to 5, where the scale attributions were: 1 - Very Bad, 2 - Bad, 3 - Fair, 4 - Good and 5 - Very Good. In these items, characteristics of the flippes classrooms were evaluated in different aspects, such as the scope of the activity, degree of difficulty, duration and if the planning and content generated so far in the subject facilitated the performance of the activity by the students. All students answered in a complete the delivered questionnaire. The results can be seen in Figure 3 and then there is a discussion of the results.

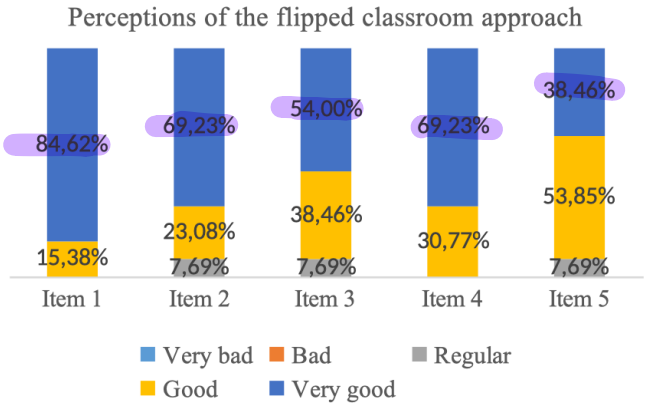


Fig. 3. Results of students' perception of the flipped classroom approach.

Regarding the scope of the activity, item 1, the students gave a very positive feedback with 84.62% of "Very Good" the construction of the flipped classroom was highly praised by the students and having a mentoring moment before the presentations was considered a success by the students. It was mentioned that for this reason they could take more risks and bring problems that they are having outside the environment of the subject. Another student commented that the flipped classroom allowed him to bring problems that he was having in no work environment and he received several potential solutions from teachers and students on other teams.

For item 2, the degree of difficulty of the activity, even with 69.23% of students responding as "Very Good", there were some points for improvement that were highlighted in the feedback, such as: (1) one student commented "*Although*

it is an exchange of experiences and learning with the expertise of students who experience the area of software engineering, the degree of difficulty of the themes was high for him who was from another area of computing knowledge", (2) another student mentioned that "*The process of having to study a subject to explain it seems to help in fixing and improving the understanding of this subject, but not having previously studied the subject of other groups ended up limiting the participation in the presentation of the other teams*". Therefore, the authors analyzed that a good practice for a future flipped classroom is for teams to make their presentations available a few days before classes, thus facilitating the discussion of students from other teams on the topic.

For item 3 on duration of activity time, 54% of students evaluating it as "Very Good" was a point of a lot of feedback with some students mentioning that "*The duration of the presentation of each team is the aspect I liked the most. I believe that in this case the time was enough for the teams to present their findings and for the audience to assimilate it without becoming tiring*". Another students mentioning "*The time for each presentation to be short*" as a suggestion several students put in increasing the time for discussion of the activity. The authors of the paper consider time to be a very sensitive characteristic to deal with and good time management is a very necessary skill in the software industry, so in future classes the authors will keep the time up to 40 minutes for presentation and 20 minutes for discussion and questions for the team.

Item 4 in relation to the knowledge acquired in the subject having helped had 69.23% with students evaluating it as "Very Good" and the rest with "Good". Students considered the knowledge acquired in previous classes as essential for the success of this activity. It was mentioned by most students that what was learned in dialogic expository classes was put into practice. One student mentioned that the flipped classroom served for "*Some students gave examples from their workplaces or market experiences on some topics they learned in class and didn't know was related to software testing*". The authors consider the feedback quite positive and consider the decision not to pass the flipped classroom in the first teaching units and with subjects that the students had not yet had any contact with.

Item 5 on the evaluation of the flipped classroom in relation to learning the content of the subject had the lowest number of responses "Very Good", a total of 38.46%, however there was no indication of "Very Bad" or "Poor" in student evaluations. In the comments of the questionnaire, the students mentioned quite positively about the activity, but that the amount of knowledge acquired with it was not compared to other approaches used, such as the Practical Project, Dojo Kake and Dojo Randori, as several points of the software testing contents need to be a more practical knowledge. Even though some students from each group made an effort to present a project in the flipped classroom, it was not always accompanied by the rest of the group, thus it may have caused a certain learning imbalance. The authors consider this feedback extremely important and very difficult to remedy in remote teaching, given that teams end up dividing functions and as the exchange of knowledge in the development of skills ends up being hampered. The next iterations of this work will be in person and this point of difficulty will be monitored and evaluated again.

V. Conclusion

This paper presented the results of a quasi-experiment of a software testing teaching approach using flipped classroom as one of the teaching approaches, taking into account that the subject was built to be supported by active teaching methodologies. Initially, the subject had been designed to be taught entirely face-to-face, but it was adapted for remote teaching and this is its second iteration.

The way to measure the results was through the analysis of the feedback collected through a questionnaire. This questionnaire evaluated all the teaching approaches used in the subject, but in this paper we analyzed only the opinion about the flipped classroom approach. students' ideas of how we can improve each of the approaches used.

Regarding the RQ presented in the introductory section "*Is it possible to guarantee engagement in a software testing subject using a flipped classroom in a remote learning scenario?*", for a satisfactory analysis of this question it was also asked in the questionnaire about whether the teaching approaches were harmed by remote teaching. In Figure 3 it can see the result of the students' answers.

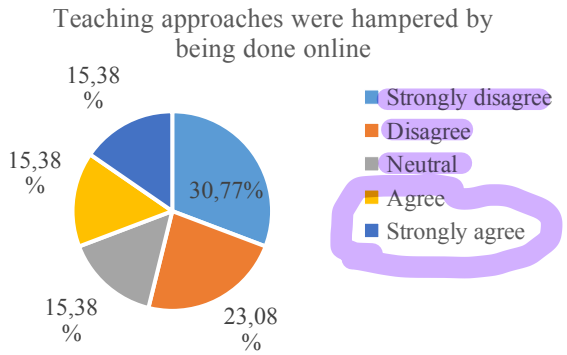


Fig. 4. Results of students' perception of the influence of remote teaching on teaching approaches.

The answer obtained from the students' perception of the flipped classroom approach in the way it was approached was quite positive in all aspects analyzed, but the opinion about remote teaching having harmed this and the other approaches used in the subject ended up having mixed answers and considerations among students. Through the students' feedback, it is seen that the exchange of knowledge between the groups in the form of remote teaching in the flipped classroom approach in the way that was planned in this paper ended up being harmed. It was suggested that more time was given to each team or that the time discussion was longer at the expense of the presentation time of each of the teams.

For the construction of the flipped classroom used, several good practices were taken into account in the use of active teaching methodologies for practical computing subjects in a remote teaching scenario [21]. For example, (1) the use of synchronous dialogued classes to help the exchange of knowledge before the flipped classroom classes, (2) the use of assistant teachers in order to serve as mentors to clarify the doubts of the teams before the presentations and (3) request that in the presentation of the teams in the flipped classroom guided examples lived by one of the team members were brought so that there was an exchange of knowledge and greater participation of the other teams in the discussion at the end of each presentation.

A. Contribuitions

The main contribution of this paper is the presentation of a way to build a flipped classroom approach that can achieve the expected results and generate skills for students of a Software Testing subject oriented to active teaching methodologies. The flipped classroom approach can be constructed in different ways [20] to achieve different expected results, so the dissemination of ways that this approach can be used is considerably relevant.

A second contribution is the indication in the face of student feedback that even with their precepts that students acquire knowledge outside the classroom initially, in the students' view the flipped classroom loses a little in being completely remote, as there is a decrease in the exchange of knowledge between teams.

A third contribution well pointed out by the students is that the mentoring classes for the teams before the flipped classrooms ended up generating a great improvement in the content presented by the teams, as seen in [2] [21].

B. Limitations

There are some limitations in this paper, any result obtained must be viewed with some caution and care must be taken in generalizing the results, one of the limitations is that the work was carried out with a graduate class in computer science, so most of the participating students already have extensive contact in Software Engineering, and some of them are also academic professors. This factor may have favored the use of the flipped classroom.

Another limitation is that the quasi-experiment was carried out with a small portion of students and was carried out in only one institution.

C. Future Works

As future works, the authors intend to replicate the flipped classroom approach in a face-to-face computer science undergraduate class, in order to verify whether some difficulties identified in the results can be avoided more easily with face-to-face teaching. However, the results collected during the quasi-experiment will already be included in the regular teaching plan of the subject, in view of the positive feedback from students about the insertion of the flipped classroom as one of the approaches, as it was not in the first iteration [21].

In the future, it is also considered to apply the approach in other computer science subjects with a less practical character to analyze whether there are visible differences in student feedback, especially the use of the flipped classroom at the beginning of the subject seeking the cognitive level of remembering from Bloom's revised taxonomy [22].

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. 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.

[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] 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.

[4] V. Garousi, M. Felderer, M. Kuhrmann, K. Herkiloğlu, and S. Eldh, “Exploring the industry’s challenges in software testing: An empirical study”. J Softw Evol Proc, vol. 32, no. 8, 2020.

[5] V. Garousi, A. Rainer, P. Lauvås jr, and A. Arcuri, “Software-testing education: A systematic literature mapping,” Journal of Systems and Software, vol. 165. Elsevier BV, p. 110570, Jul. 2020 [Online]. Available: http://dx.doi.org/10.1016/j.jss.2020.110570.

[6] E. Van Veenendaal. “Next-Generation Software Testers: Broaden or Specialize! The Future of Software Quality Assurance". Springer International Publishing. 229–243. 2019.

[7] K. Van Dam. The Future of Testing. The Future of Software Quality Assurance. Springer International Publishing. 197–205. 2019.

[8] CC2020 Task Force, “Computing Curricula 2020.” ACM, Nov. 15, 2020. doi: 10.1145/3467967.

[9] M. L. Fioravanti and E. F. Barbosa, "Outlining Software Project Management Education by Surveying Educators," 2019 IEEE Frontiers in Education Conference (FIE), 2019, pp. 1-8, doi: 10.1109/FIE43999.2019.9028366.

[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] V. Garousi, S. Bauer, and M. Felderer, “NLP-assisted software testing: A systematic mapping of the literature,” Information and Software Technology, vol. 126. Elsevier BV, p. 106321, Oct. 2020 [Online]. Available: http://dx.doi.org/10.1016/j.infsof.2020.106321

[12] Y. Zhao, Y. Hu and J. Gong, "Research on International Standardization of Software Quality and Software Testing," 2021 IEEE/ACIS 20th International Fall Conference on Computer and Information Science (ICIS Fall), 2021, pp. 56-62, doi: 10.1109/ICISFall51598.2021.9627426.

[13] ISTQB (2018). International software testing qualifications board. available at: https://www.istqb.org/.

[14] C. Raksawat and P. Charoenporn, “Software Testing System Development Based on ISO 29119,” Journal of Advances in Information Technology, vol. 12, no. 2. Engineering and Technology Publishing, pp. 128–134, 2021 [Online]. Available: http://dx.doi.org/10.12720/jait.12.2.128-134

[15] TMMI Foundation (2018). Test Maturity Model Integration – TMMI Release 1.0.

[16] 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.

[17] M. J. Lage, G. Platt, and M. Treglia, “Inverting the classroom: A gateway to creating an inclusive learning environment,” The Journal of Economic Education, vol. 31, no. 1, pp. 30–43, 2000.

[18] Y. Qu, R. Cai and M. Haj-Hussein, "Research and Practice of Applying Adaptive Learning in Computer Science and IT Degree Programs," 2019 IEEE Frontiers in Education Conference (FIE), 2019, pp. 1-8, doi: 10.1109/FIE43999.2019.9028374.

[19] S. Mithun, M. Vickery and X. Luo, "Evaluating Factors for Effective Flipped Classroom Instruction in an Advanced Data Management Course," 2021 IEEE Frontiers in Education Conference (FIE), 2021, pp. 1-9, doi: 10.1109/FIE49875.2021.9637247.

[20] J. A. De Sena Quaresma and S. R. B. Oliveira, "Teaching and learning strategies for software process subject," 2021 IEEE Frontiers in Education Conference (FIE), 2021, pp. 1-7, doi: 10.1109/FIE49875.2021.9637428.

[21] 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.

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

[23] L. Cronbach, “Coefficient alpha and the internal structure of tests”. Psychometrika, n. 16, 1951.