A diagnosis on software testing education in the Brazilian Universities

Isaac Souza Elgrably
Graduate Program in Computer Sciense (PPGCC), Institute of Exact and Natural Sciences (ICEN)
Federal University of Pará (UFPA)
Belém, Pará, Brazil
isaacelgrably@gmail.com

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

***Abstract*— This Research to Practice Full Paper presents an investigation on the teaching of software tests in Computer Science courses in Brazilian Institutions. In this work, we investigate the existence of subjects and contents related to the software testing theme in the academic curricula of the best Brazilian universities in the computer courses indicated by Folha's Universities Ranking in 2019. So that the software testing activity becomes widespread and recognizable, it needs to be widely taught and discussed, although there is a growing concern with this teaching theme, the organization and depth of teaching software testing varies widely in each curriculum matrix of the studied institutions, especially comparing them with the curriculum of updated reference of the ACM / IEEE (Association for Computing Machinery / Institute of Electrical and Electronic Engineers) and the curriculum guide of the SBC (Brazilian Computer Society) along with the areas of knowledge present in the SWEBOK Guide (Software Engineering Body of Knowledge), which are the main references for building of the curricula of Brazilian institutions. Thus, this work provides important inferences about how software testing has been approached, at the curricular level, in different Brazilian institutions, from specific subjetcs of software testing, subjects that contain topics related to tests and which are the main topics that are taught. Professors and academic managers can use the knowledge generated in this paper to help in a possible readjustment of how their institutions can deal with the teaching of software tests.**

Keywords—computer science, syllabus, software testing.

I. INTRODUCTION

Taking into account the report on the construction of the new ACM / IEEE curriculum guide, for the Computing Curricula 2020 Project [1], which had as one of its main objectives to evaluate the world state of computing education. This work provides a diagnosis on the Software Testing or can also be referred to as the area of computing knowledge 4.1. Software Quality, Verification and Validation [1].

When software does not work properly, it can cause many problems, including loss of money, time or business reputation, and even injury or death to humans or animals. Software testing aims evaluating the software quality and reducing the risk of failure of the software in operation [2].

According to the TMMi model (Test Maturity Model Integration) [3], the importance of the test subject, as one of the quality measures that can be taken, is growing rapidly. Testing has become a key activity that directly influences not only the product quality, but also the "performance" of the entire development and manufacturing process.

Analyzing that the activities related to the test generally correspond to 30 to 40% of the total costs of the project, only limited attention is still given to it [3]. Thus, the need for better use in learning this important topic of computing is consolidated.

The purpose of this diagnosis is to find out how the teaching of software testing in computer science courses at Brazilian institutions is progressing. It was identified how the institutions are dealing with the need to convert it into a subject, the way it is partially taught in other subjects and the main contents contained in the academic curriculums of each institution were verified.

To build this test diagnosis, a search was made in the specialized literature with an emphasis on published studies. The main related work was the diagnosis of 2015 [4] that served as inspiration for the construction of the selection methodology of educational institutions. However, the main object of study is different, its focus was to present software testing education initiatives, educational trends and how games support the education of Software Testing.

This work presented was focused on verifying, analyzing and correlating the contents that are taught about Software Testing in the institutions. The test curriculum model [5] was used in this research as a point of comparison in conjunction with the curriculum guide of the ACM / IEEE [6] for delivering a test curriculum model taking into account the Brazilian curriculum guidelines [7], which it made possible a point closer to the teaching competences that are indicated for Brazilian institutions.

In addition to this introductory section, this paper is structured as follows: in section II a theoretical basis is provided on the precepts used in this work, in section III the research methodology is described, showing how it was organized and what its objectives were in addition to the results obtained, in Section IV a comprehensive view on the teaching of software testing is presented, in Section V a discussion of the teaching of software testing in Brazilian institutions is presented, in Section VI we conclude this work, presenting the contributions of this model, in addition to present its limitations and highlight the next steps for the continuation of this research.

II. BACKGROUND

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

A. Software Testing Education

Tests are one of the most valued programming practices in any type of software. It is also one of the practices that has undergone the most changes in recent years [8].

There is a need to try new pedagogical models in the teaching of software testing, especially those that seek to promote greater conditions for the student to develop

2021 IEEE Frontiers in Education Conference (FIE) | 978-1-6654-3851-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/FIE49875.2021.9637305

autonomy, critical thinking and at the same time that the student is able to have a good theoretical-practical training, aligning the state of art with emerging technologies in the field [9]. There is also a need to invest in the academic education of computing students. Computing students are expected to incorporate software testing concepts and practices through the development of initiatives [10].

In defining topics to be taught, software testing is often overlooked in favor of design and implementation activities [11]. Several studies have suggested changes and improvements in relation to software testing: suggestions for building a test curriculum [5], use of methodology aimed at supporting the teaching of software testing [11, 12, 13], studies on content related to tests [14, 15].

So, in spite of verifying in the literature the importance and the changes that have been occurring in software testing, the authors of this work sought to find out how Brazilian educational institutions have been providing options for teaching and learning regarding the content of software testing. Since the verification of how software testing is being taught within institutions is a point little addressed by researchers in the area of software testing.

B. SBC and ACM Curriculum for Computing

The Brazilian Computer Society in 2017 prepared "computer training references" for each of the courses contained in the National Curriculum Guidelines at MEC [7]: Computer Science, Computer Engineering, Software Engineering, Teaching in Computing and Systems Information. The entity informs that they are not curricula, but consultation material for those who are preparing their curricula. This material must be worked in conjunction with the guidelines of the Brazilian Ministry of Education [16].

Specifically in the material of the bachelor's degree in computer science, the presence of Software Testing content is based on the formation axis of systems developments and the knowledge about software testing is contained in the Software Engineering content.

The ACM / IEEE reference curriculum [17] defines a structure of topics and units of knowledge in order to specify teaching and professional training guidelines for the study areas of computing. It serves as a teaching base for computing topics for several universities, being recognized and adopted internationally, being widely used in conjunction with other guides for curriculum construction worldwide.

In the ACM / IEEE reference curriculum [17] software testing is included entirely in Software Verification and Validation and partially in Software Construction and Software Design [18]. The curricular guidelines present in the ACM / IEEE guide [6] are widely used for the construction and maintenance of undergraduate curricula on subjects related to Software Engineering. The ACM / IEEE guide [6] describes the main core for a software engineering curriculum, it defines a set of content for each subject to be addressed in the curriculum [19].

It is important to note that the new ACM / IEEE curriculum is under construction, seeking to enhance the students' learning environment and build knowledge via necessary skills and competences for the software industry [1, 20, 21].

1) Software Testing content present in Curricula: Among the possibilities of Software Testing content guides we have the main references in the SWEBOK [22] curriculum guides, the ACM / IEEE [6, 17] and SBC Guides [16, 23]. However, there are other works that also offer specialized academic curricula for Software Testing as a test curriculum model [5]. For the evaluation of content found in Brazilian institutions, the proposals of the ACM / IEEE Guide [6] were selected, which is serving as a reference for the construction of the knowledge areas of the Curricula 2020 Project [1] and the test curriculum model [5], which proposed a curriculum taking into account the context of Brazilian education.

Then, the knowledge areas, knowledge units and topics for teaching suggested by the ACM / IEEE guide [6] present teaching units and topics on Software Verification and Validation and suggest for each teaching topic which Bloom's taxonomy [24], which will be used as a basis for comparison in Subsections C and D of Section IV of this work. It is important to point out that the Institutions are not obliged to use a curriculum guide to build their curriculum, but it is important to check the compatibility to find out what is being taught in the content of Software Testing.

As mentioned, the contents will also be compared with a curriculum created specifically to support the teaching of Software Testing in the context of Brazilian teaching from the works [5, 18]. The curriculum focused on the teaching of testing described in [5], which includes the teaching of Software Testing from the knowledge areas of Software Engineering, Software Construction, Software Quality and Testing and to achieve the expected results, the revised Bloom's taxonomy [25] is used for each content indicated by the guide. The content of this curriculum will be compared to the curricular matrices of the Institutions in Subsection B of Section V in order to verify the adherence between the subjects of the computer science courses of the Brazilian institutions.

III. RESEARCH METHODOLOGY

For the selection of Brazilian teaching institutions that had their curricula analyzed, it was based on the proposal made by Valle, Barbosa and Maldonado [4], which aimed to identify how the teaching of software tests in computer science undergraduate courses in Brazilian and foreign institutions had been approached through a curriculum analysis, with the difference that it was decided to use the specific ranking for computing courses, so the 2019 RUF (Ranking University Leaf) the most current until the writing of this paper.The ranking of RUF courses is an annual evaluation of graduations from universities, university centers and colleges in Brazil, using national and international data from opinion polls in two aspects: teaching and the market.

From the ranking orders, it was verified the grade attributed by the Brazilian Ministry of Education to see if there was a conformity with the ranking grade and if the institutions had a computer science course, which was the object selected for this research because these courses have already been regulated in Brazil by SBC since the mid-1980s [15], in addition to being the most present in the best placed institutions in the RUF. And, finally, the accessibility of the curricular matrices was verified.

The intention was to expand the selection of institutions in the work of Valle, Barbosa and Maldonado [4] from 25 to 32 Brazilian institutions, however 4 instutions did not pass the established criteria. Thus, a total of 28 institutions, the curricula, pedagogical plans and syllabus of each one of these institutions were collected and for each subject content related to software testing was collected, this will be better explained in Section IV of this work.

This research gathered the content of the curricular matrices of the main teaching institutions of computer courses in Brazil. From the analysis of the curricular matrices, it was possible to generate a framework of knowledge that was used to reach some conclusions that can be verified in the following research questions: (RQ I) What is the incidence of Software Testing subjects in computer science courses at the main Brazilian institutions? (RQ II) Is software testing being taught in computer science courses in other subjects at major Brazilian institutions? (RQ III) How is Software Testing being taught in computer science courses at major Brazilian institutions?

Regarding RQ I, its objective is to verify if the topic of software testing has already been considered relevant enough that Brazilian educational institutions in their curricular matrices have inserted specific subjects on the subject. For this, the authors made an analysis on teaching plans, syllabus or curricular matrices available in the institutions' website in order to obtain the number of subjects specifically on the software testing, in addition to all their academic content. Then, the results were presented to another software engineering researcher who has experience in creating academic curricula and teaching plans for evaluation and considerations of the results obtained.

For verification of RQ II, the same analysis was done as in RQ I, but accessing the content of each of the subjects present in the curricular matrices of each of the institutions and it was verified if topics of software testing are being taught, besides to check which subjects are the most used to transmit the content related to software testing. Similar to RQ I, all the results obtained were evaluated by another researcher in the Software Engineering.

For RQ III, the curricular matrix of each of the institutions was analyzed in full, in order to find out how the institutions are teaching software testing. A diagnosis was made of what is contained in the curricula of each of the institutions in relation to the topics of software testing obtained in the Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering [6] and in a test curriculum model [5].

IV. Diagnosis of Education in Software Testing in the Brazilian Scenario

From the selection presented in Section III, the curricula of the national institutions were analyzed, thus building a diagnosis of how the institutions are delimiting teaching content on software testing and what the main knowledge these undergraduate students bring to the job market.

A. The Brazilian Institutions considered

The teaching institutions that the most relevant Computer Science curricula in Brazil according to the ranking of courses made available by the RUF in 2019, can be seen in Table I in ascending order of classification.

Of these 28 main Brazilian institutions that were analyzed, there were some ones that were ahead of these, but did not meet the other criteria, such as the Technological Institute of Aeronautics - ITA that does not have a computer science course and the Pontifical Catholic University of Paraná - PUC PR and University of Vale do Rio dos Sinos - UNISINOS that did not have available curricula, pedagogical plans or syllabuses.

TABLE I. Brazilian Education Institutions considered

Brazilian Universities	Abbreviation
State University of Campinas	UNICAMP
Federal University of Minas Gerais	UFMG
Federal University of Rio Grande do Sul	UFRGS
Federal University of Rio de Janeiro	UFRJ
Federal University of Pernambuco	UFPE
University of São Paulo	USP
Pontifical Catholic University of Rio de Janeiro	PUC Rio
Federal University of São Carlos	UFSCAR
Federal University of Paraná	UFPR
Federal University of Santa Catarina	UFSC
State University of Paulista Júlio de Mesquita Filho	UNESP
Pontifical Catholic University of Minas Gerais	PUC Minas
Federal University of Campina Grande	UFCG
Federal University of Technology – Paraná	UTFPR
Pontifical Catholic University of Rio Grande do Sul	PUC Rs
Federal University of Ceará	UFC
Federal University of Bahia	UFBA
Federal University of Uberlândia	UFU
Federal University of Amazonas	UFAM
Mackenzie Presbyterian University	MACKENZIE
University Center of Educational Foundation of Ignatius Father Sabóia de Medeiros	FEI
Fundação Federal University of Abc	UFABC
University of Brasília	UNB
Federal University of Rio Grande do Norte	UFRN
Federal University of Alagoas	UFAL
Federal University of Goiás	UFG
State University of Rio de Janeiro	UERJ
Federal University of Pará	UFPA

In all institutions, subjects were found covering software testing content. The main way in which Brazilian institutions approach software testing content is from the subjects of Software Engineering and Software Quality. In this case, generally, only a part of the course is used to present content about software testing.

B. Existing Software Testing Subjects

A considerable evolution was verified in the number of subjects specifically destined to the teaching of Software Testing in the institutions, verified from the work [6], as can be seen in Figure 1.

Of the 19 institutions that have software testing subjects, only in two this subjects is mandatory: Pontifical Catholic University of Rio de Janeiro and University Center of Educational Foundation of Ignatius Father Sabóia de Medeiros. Among the institutions analyzed, the University of São Paulo, the Federal University of Rio Grande do Norte and the Federal University of Alagoas stood out, which have in their curricular matrices two subjects aimed at teaching the content of Software Testing.

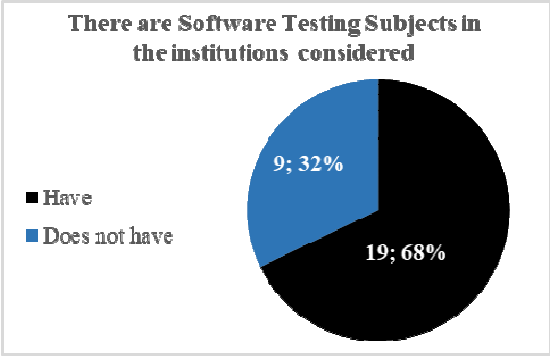

Fig. 1. Institutions that have Software Testing subjects

Another parameter that was identified in the testing diagnosis is that there is no consensus on the content or even the name of the subject for software testing. In Table II it can see the different names of subjects for software testing found in the institutions considered.

TABLE II. OCCURRENCES OF SOFTWARE TESTING SUBJECTS

Subject name	Occurrences
Software Testing	7
Software Verification and Validation	3
Software Validation	2
Software Verification	2
Software Test Project	1
Software Simulation and Testing	1
Model-Based Software Testing	1
Software Testing and Inspection	1
Software Measurement and Testing	1
Software Testing II	1
Software Testing Topics	1
Software Verification, Validation and Testing	1

From the results obtained by the diagnosis, it is possible to see that there is still no standardization for the name of a subject in the Software Testing content. It was also found that there was a tendency in the institutions to merge the content of Software Testing with other topics of Software Engineering, such as Software Measurement and Inspection.

Another verified trend was to integrate the topic of Software Testing in a practical subject as seen in the Software Testing Project and Model-Based Software Testing subjects, which have in their teaching plan an activity of delivering a software product, which complies with the indication given by the ACM / IEEE guide [20]. Finally, at the University Center of Educational Foundation of Ignatius Father Sabóia de Medeiros there is a subject called Software Simulation and Testing that focuses on teaching concepts of Computer Theory in conjunction with the practice of the type of structural test or white box, by ISTQB (International Software Testing Qualifications Board) definition [2].

Another parameter considered important by the researchers on these subjects found in the diagnosis, were the main contents addressed in each of them, which can be seen in Figure 2 and explained below.

The contents were adjusted and grouped for better consideration, for example: Test Fundamentals, Initial Concepts, Basic Concepts were grouped into Basic Test Concepts. The large number of Test Techniques is due to the fact that several curricular matrices enumerate the test techniques that will be in the content. Test Driven Development [26] is a test practice that most appears in the content in a specific way, a practice widely used for teaching Software Testing and Programming, as seen in [12, 27].

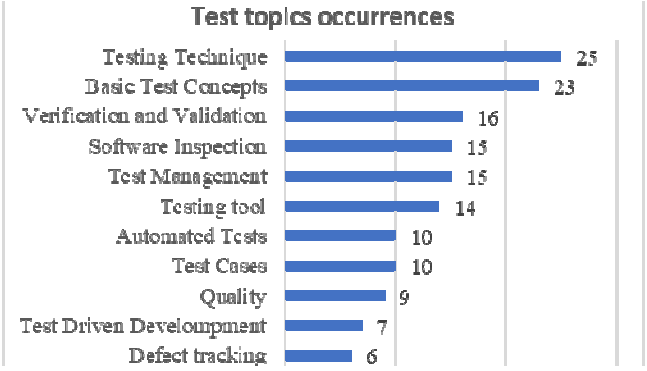

Fig. 2. Subjects most present in curricula

It is important to state that different institutions treat their curricular matrices differently, without any normalization, for example: the curriculum of the Federal University of Alagoas had only one content "Study of static and dynamic techniques for software verification" while the curriculum of the Federal University of Ceará had twenty-two contents in the curriculum matrix. Only with inferences from what was found in the curricular matrices, it is not possible to affirm the entire scope of the teaching of Software Testing in each of the Institutions.

C. Software Testing Topics included in other Subjects

Regarding subjects that have Software Testing topics, all verified institutions have at least one subject with these topics. The topic of testing was considered to be content containing topics in the Software Verification and Validation knowledge area of the ACM / IEEE Guide [6]. The number of subjects per institution can be seen in Figure 3.

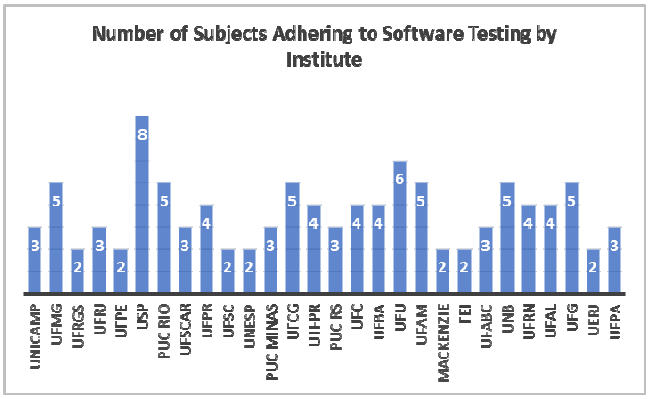

Fig. 3. Number of subjects per institution

On Figure 3, it is possible to see a large discrepancy between the number of subjects that present topics adhering to software testing in different Brazilian institutions. The authors consider that this may be due to different factors, such as: recent updating of the curricular matrices of institutions, researchers and experts in software testing among the institutions' professors, proximity of institutions to the software industry and organization of the content made available in the their curricular matrices, teaching plans or syllabuses.

According to the ACM / IEEE Guide [17] the topic of Software Verification and Validation is part of the subject of Software Engineering, so it and other subjects usually have content related to Software Testing. One of the results of this diagnosis is the number of subjects that address topics related to Software Testing in the institutions considered. This value and its frequency of appearances can be seen in Figure 4.

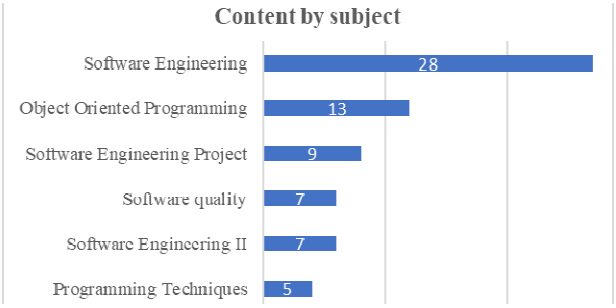

Fig. 4. Main subjects with Software Testing content

The names of some subjects were adjusted and grouped for better consideration, for example: Software Engineering I, Principles of Software Engineering and Fundamentals of Software Engineering were grouped into Software Engineering. Some perceptions found are that in the object-oriented programming and programming techniques subjects, the contents that are given are related only in the software verification stage and mainly using unit tests and software inspection. In the subject of Information Systems Design the focus is on acceptance tests. An interesting data collected is that the Quality subjects do not always have content from Software Testing, so only the subjects that had this content were considered.

Finally, some other perceptions reached were that all 28 institutions analyzed have software testing content in their curricular matrices in at least more than one subject. Some subject of Software Engineering is always mandatory in each of the institutions and the main contents in these subjects regarding tests are: software testing, software verification and software validation.

D. Relationship between the Teaching of Software Testing and Curricula in the Institutions considered

In this topic of the diagnosis, the comparison between the content indicated by the ACM / IEEE Guide [6] and the curricular matrices of the institutions that are being analyzed in this work will be presented. As the objective is to make a diagnosis of software testing at the level of institutions, it was decided to compare the content of software testing of all the subjects present in the computer science courses of each institution with the ACM / IEEE Guide [6]. Then, in Figure 5 the degree of compatibility between the contents present in the ACM / IEEE Guide [6] and the average of the contents found in the subjects of the analyzed institutions can be seen.

The topics most present in the subjects with test content are from the "Testing" knowledge unit, among them the highlights for Black-box functional testing techniques that are present in 86% of the studied curricular matrices. ISO / IEC / IEEE 29119 [28] specifies that black-box tests are based on external inputs and outputs for a test item, usually based on a specification, rather than its implementation in code or software. Thus, black-box testing can be approached at different levels and types of tests, facilitating its adaptation to the objectives of the curricular matrices of the institutions.

In contrast, the topic of the ACM / IEEE Guide [6] with less presence in the curricular matrices is Exception Handling, with only 18% presence even though it is an important topic to be used in conjunction with unit tests, coverage analysis and for making data and test cases.

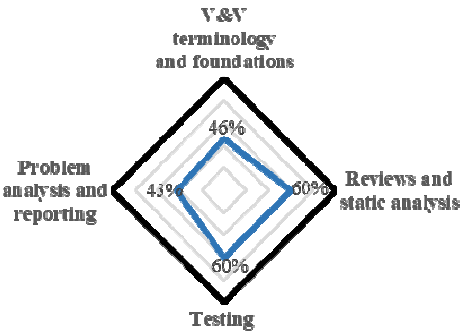

Fig. 5. Compatibility between the topics of the Knowledge Units [4] with the contents of the considered Institutions.

Regarding the issue of institutions, the Federal University of Ceará is the one that has the greatest compatibility with the ACM / IEEE Guide [6], 82% of all topics present in the guide are in the contents made available by the Institution around five subjects: Verification and Validation, Software Engineering, Software Quality, Software Factory and Requirements Engineering. However, it is also important to show that of these subjects only that of Software Engineering is mandatory, all others are optional. So, even with good compatibility, not all students will have access to all the highlighted knowledge.

Finally, there was a great discrepancy between the topics of the Software Testing content present in the main Brazilian institutions with computer science courses, this may be due to the fact that in several of these institutions the curricular matrices are outdated, with more than five years without updates and most of the new updates are due to the insertion of optional subjects.

E. Additional Items included in the Teaching of Software Testing and which are not part of the Curricula in the Institutions considered

During the diagnosis, these items were separated for analysis by the researchers, however, the content of the topics present in the twenty-two Software Testing subjects was only specifically analyzed to ensure that the content was thought to support the teaching of tests.

One of the topics found was code smells, which according to Valente [8], are indicators of low quality code, that is, code that is difficult to maintain, understand, modify or test. This principle is important for its relationship with unit tests, automated tests, code coverage and defect analysis. Another additional item, this widely verified in several works [29, 30, 31], the analysis of mutants or mutant tests, which are widely used to ensure the integrity of code coverage and are very efficient if used with automated tests.

Some curriculum matrices also indicate the teaching of Risk-Based Testing content, which can be used to optimize existing testing activities, introducing a risk-based strategy for prioritization, automation, selection, resource planning and so on [32] and can be used to reduce the appearance of defects in the long run [33]. Other topics are related to software testing in the view of quality models such as TMMi [3], CMMI (Capability Maturity Model Integration) [34] and

in the view of ISO / IEC / IEEE 29119 [28]. This relationship on the content of tests in these parameters is beneficial to students to present challenges that they may face in the job market.

V. Discussion

In this section, we discuss the main discoveries made in this work, in addition to considering the reason for choosing to use a diagnosis in curricular matrices as a research methodology.

A. Closing Analysis

The choice of the methodology of making a diagnosis in the curricular matrices of the institutions instead of, for example, conducting a systematic review of the literature [35] comes from the fact that the researchers wanted to verify what is being taught specifically in the undergraduate courses in computer science in Brazilian institutions, while in a review researchers and case studies could be reached in subjects delimited in a diagnosis, it is possible to verify the way that the contents of software testing have been occurring, mainly at a time of construction of a new ACM / IEEE curricular guide.

Through the results obtained in Section IV, we can see that Software Testing already has a considerable degree of importance among the educational institutions considered. In Figure 1 we see that 68% of the institutions have at least one specific subject for software testing, however the vast majority of them are optional, in which the student has the option or not to participate in the subject.

Another verified point is that the amount of content taught about Software Testing varies a lot in the institutions. There are some like the Federal University of São Paulo that presents test topics in ten subjects and others like the State University of Paulista Júlio de Mesquita Filho that has only test topics in two subjects.

In all selected institutions, there is a Software Engineering subject and it has at least one Software Testing topic in its content. Another parameter found is that several Algorithms and Programming subjects from the first semesters have testing and debugging topics.

Regarding the compatibility between the curricular matrices and the ACM / IEEE guide [6], the knowledge unit that is least covered by the institutions is the Problem analysis and reporting that deals with failure reports, defect analysis and problem tracking, some these topics were found in Devop's subjects and in Agile Methods Laboratories, but in no subject of Software Engineering or Software Quality. Defect analysis is of great importance for the maintainability and evolution of the software, the ISTQB [2] informs that it is a technique dominated by more experienced testers, which may be the reason for its little presence in the content of the analyzed institutions.

Another parameter verified in the curricular matrices is that they indicate for the teachers to carry out practical projects [36, 37, 16] and use active methodologies to teach the student topics related to software testing [12]. Finally, it was found that the institutions also add new items to the content of software testing, such as: quality models, agile methods or mutant tests; and they can diversify the knowledge of the area.

B. Analysis of a Curriculum Model of Software Testing in relation to those of the Institutions considered

Another analysis carried out was to make a comparison between the contents of Software Testing of the curriculum model [5] and the curricular matrices of the institutions that are being analyzed in this work.

This choice was due to the curriculum model presenting a separation of Teaching Units based on a mapping [18] with the ACM / IEEE [17] SBC [16, 23] curricular guides, the SWEBOK knowledge guide [22], the quality model of TMMi [3] and agile principles and practices [38, 39]. As it is a recent work and not having used the ACM / IEEE Guide [6] for its construction, the authors considered it interesting to complement the diagnosis by comparing its suitability to Brazilian educational institutions.

This adaptation was made in two scenarios: the first similar to that carried out in Section IV comparing the curriculum model with all the contents of Software Testing of each institution, whereas the second scenario took into account only the subjects of Software Testing of the institutions. If the same institution has more than one subject related to Software Testing, the non-repeated content of the subjects was taken and thus the comparison was made. In Figure 6 it can see the percentage of compatibility of the average of all content found in the subjects of the institutions related to tests.

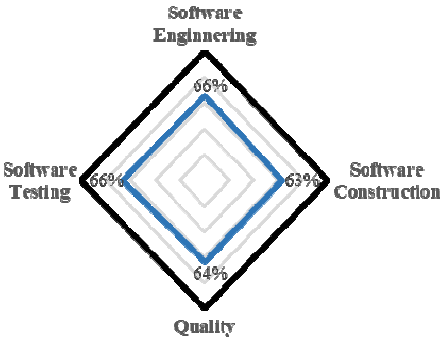

Fig. 6. Compatibility between the Test Curriculum Model [5] with the contents of the Institutions considered

The compatibility of the test curriculum mode with the average of the institutions was 66% for each teaching unit, which is already quite satisfactory. The curriculum to have content related to subjects as teaching units can be considered one of the reasons, in addition to the fact that the curriculum uses SBC guides [16, 23] and SWEBOK topics, which are constantly used to create learning.

Unlike the analysis made with a teaching unit that highlights the topics present in the curriculum model, they end up having considerable proximity. However, a topic is very comprehensive in all the curricular matrices analyzed it is the “Software testing techniques”, which has a 92% compatibility with the curricula considered. The ISTQB syllabus [2] considers that testing techniques are considered three different groups: white-box techniques, black-box techniques and experience-based techniques, which makes this topic cover a large amount of content. On the other hand, the least present topic was “Improvements in the testing process” with a compatibility of 14%, which may be due to the fact that few institutions make the interaction of Software Testing with quality models.

On the issue of institutions, the Federal University of Rio Grande do Norte has 96% compatibility with the test curriculum model mentioned through the content present in six subjects: Software Testing, Software Testing II, Software Engineering, Software Quality, Good Practices for Programming, and Systems Specification and Verification. Of those mentioned, only the subjects of Software Engineering and Systems Specification and Verification are mandatory. In Figure 7 it can see the percentage of compatibility of the average of all content found in the subjects of the institutions only of the specific subjects of Software Testing.

The compatibility of the test curriculum model with the average of the testing subjects present in the institutions is 68%, this growth considered the result of the compatibility with all curricular matrices is due to the fact that only the 19 institutions that have Software Testing subjects have been analyzed for that scenario. However, with the exclusivity of the comparison with only specific subjects, there was a decrease in compatibility in other teaching units, which shows us that the content related to Software Testing in the considered institutions ends up being delivered in a multidisciplinary way.

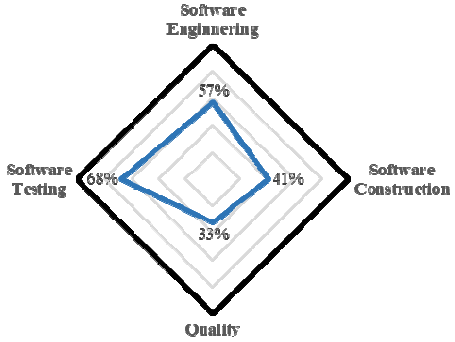

Fig. 7. Compatibility between the Test Curriculum Model [5] with the Software Testing subjects of the Institutions considered

In the second scenario there were some topics with 100% presence in all the institutions considered, they are “Testing fundamentals”, “Software testing techniques” and “Approaches to verification”. These topics can all be approached in a more classic way of teaching or with more active methodologies, can also compose practical projects, besides being very relevant for the evolution of the knowledge of the contents of Testing.

The Federal University of Rio Grande do Norte with its two subjects of Software Testing has a compatibility of 78% with this curriculum, very close to UNB with a subject of Software Testing has a compatibility of 71% with the curriculum. This shows that the topics indicated in this curriculum model are already present in the institutions, but still far from the totality. It is also important to indicate that no topic in this curriculum is not compatible with the Software Testing subjects of the institutions considered.

C. Limitations

There are some limitations in this research, the main one being that the curricula analyzed were all from Brazilian institutions, which delimits the results obtained. Another limitation is on the curricular matrices that were used, the most current one available on the institutions' platforms was selected, in addition to comments and pedagogical projects. A final threat was related to the conversion of the content terms found, while some institutions covered as much information as possible in their curricular matrices, others left vague terms in the teaching topics.

VI. Conclusion

The objective of this research is to contribute to the development of the test teaching in the Brazilian scenario, presenting a diagnosis with the contents of software testing that the main Brazilian institutions currently have. The diagnosis presented a set of results that were analyzed and discussed by different researchers, results such as: number of software testing subjects in the institutions considered, main subjects with Software Testing content, alignment with curricula and content that are being taught in Software Testing subjects.

With regard to the research questions raised for this work, on the RQ I the answer is relatively positive, the incidence of Software Testing subjects is part of 68% of the institutions considered, in addition, as already mentioned, there are institutions that have more than a software testing subject, these data already show a trend towards the consolidation of Software Testing as, possibly, a more common subject in the Brazilian teaching scenario.

As for RQ II, the answer is positive, due to Software Testing topics are being taught in at least one subject in all institutions considered in some cases in eight different subjects. On this issue, it was also verified that test topics are present in several subjects recently included in the institutions and in subjects that revolve around practical projects for the design of a product. Another fact that can also be highlighted is that we are already seeing the introduction of software testing topics in subjects from the first semesters of the considered institutions, such as Algorithms and Computer Programming, which already has unit testing topics in its content and code debugging and analysis.

For RQ III, the result was considered relevant because it was seen that the curricula of the institutions do not have an alignment with a single source of reference and that some institutions have content that the curricula used as a reference do not have. The reason may be that in in recent years there has been an increase in the interest of researchers in relation to the teaching process of learning content related to software testing [4].

The authors clarify that this work does not aim to criticize the curricular matrices of Brazilian educational institutions, it sought to provide points for improvement and adaptation of curricula and support for content and references that can be used in a curricular construction.

A. Future Works

It is expected to replicate the methodology by analyzing the teaching of software testing topics at the main universities in the world. It is also intended to replicate the study with other Brazilian technological courses indicated by the SBC [20] and to analyze the similarities and discrepancies in the teaching of software testing between them. Finally, in the future after the consolidation of the Computing Curricula 2020 Project [1], it is expected to redo this research by updating to the new content related to tests and verifying the acceptability of the new curricular guide in Brazilian institutions.

ACKNOWLEDGEMENTS

The authors would like to thank CAPES (Coordination and Improvement of Higher Level or Education Personnel) for granting an institutional Doctoral scholarship to the author of this work. This work belongs to SPIDER Project (http://spider.ufpa.br).

REFERENCES

[1] S. Frezza, T. Clear, and A. Clear, "Unpacking Dispositions in the CC2020 Computing Curriculum Overview Report". 2020 IEEE Frontiers in Education Conference (FIE), Uppsala, Sweden, 2020, pp. 1-8, doi: 10.1109/FIE44824.2020.9273973.

[2] ISTQB, International software testing qualifications board. 2018, available in: https://www.istqb.org/.

[3] TMMI Foundation, "Test Maturity Model Integration – TMMi". Release 1.0, 2018.

[4] P. H. D. Valle, E. F. Barbosa, and J. C. Maldonado, "Cs curricula of the most relevant universities in brazil and abroad: perspective of software testing education". In 2015 International Symposium on Computers in Education (SIIE), 62–68, 2015.

[5] I. S. Elgrably and S. R. B. Oliveira, "Construction of a syllabus adhering to the teaching of software testing using agile practices". 2020 IEEE Frontiers in Education Conference (FIE), Uppsala, Sweden, 2020, pp. 1-9, doi: 10.1109/FIE44824.2020.9274266.

[6] M. Ardis, D. Budgen, G. W. Hislop, J. Offutt, M. Sebern, and W. Visser, "SE 2014: Curriculum guidelines for undergraduate degree programs in software engineering". Computer, no. 11, pp. 106-109, 2015.

[7] MEC, National Curriculum Guidelines for Undergraduate Computing Courses (DCN16). Brazil. 2016.

[8] M. T. Valente, "Modern Software Engineering: Principles and Practices for Productive Software Development". 2020

[9] L. N. Paschoal, "Contributions to Teaching Software Testing with the Flipped Classroom Model and a Conversational Agent". Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP. 2019.

[10] L. N. Paschoal, M. M. Oliveira, S. M. Melo, E. F. Barbosa, and S. R. S. Souza, "Evaluating the impact of Software Testing Education through the Flipped Classroom Model in deriving Test Requirements". presented at the SBES '20: 34th Brazilian Symposium on Software Engineering, Oct. 2020, doi: 10.1145/3422392.3422489.

[11] G. M. de Jesus, F. C. Ferrari, L. N. Paschoal, S. do R. S. de Souza, D. de P. Porto, and V. H. S. Durelli, "Is It Worth Using Gamification on Software Testing Education? An Extended Experience Report in the Context of Undergraduate Students". JSERD, vol. 8, pp. 6:1 – 6:19, Aug. 2020.

[12] 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), Uppsala, Sweden, 2020, pp. 1-9, doi: 10.1109/FIE44824.2020.9274117.

[13] L. P. Scatalon, R. E. Garcia, and E. F. Barbosa, "Teaching Practices of Software Testing in Programming Education". 2020 IEEE Frontiers in Education Conference (FIE), Uppsala, Sweden, 2020, pp. 1-9, doi: 10.1109/FIE44824.2020.9274256.

[14] M. A. Umar, "A Study of Software Testing: Categories, Levels, Techniques, and Types". TechRxiv. 2020. Preprint. https://doi.org/10.36227/techrxiv.12578714.v1

[15] G. Tebes, D. Peppino, P. Becker, G. Matturro, M. Solari, and L. Olsina, "Analyzing and documenting the systematic review results of software testing ontologies". Information and Software Technology, vol. 123, p. 106298, Jul. 2020, doi: 10.1016/j.infsof.2020.106298.

[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). 153p, 2017. ISBN 978-85-7669-424-3

[17] ACM/IEEE, "Computer science curricula 2013". Curriculum guidelines for undergraduate degree programs in Computer Science. 2013.

[18] I. Elgrably and S. Oliveira, "A Proposal for Teaching and Applying Focused of Agile Tests Methods made by an Assets Mapping". 16th International Conference on Information Systems & Techology Management – CONTECSI, 2019.

[19] D. Budgen, P. Brereton and N. Williams and S. Drummond, "What support do systematic reviews provide for evidence-informed teaching about software engineering practice?". e-informatica software engineering journal., 14 (1). pp. 7-60, 2020.

[20] B. V. M. Sabin, H. Alrumaih, J. Impagliazzo, B Lunt, M Zhang, B Byers, W Newhouse, B Paterson, S Peltsverger, and C Tang, "Information Technology Curricula 2017: Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology A Report in the Computing Curricula Series Task Group on Information Technology Curricula". ACM, 2017.

[21] L. Waguespack, H. Toppi, S. Frezza, J. Babb, L. Marshall, S. Takada, G. Veer, and A. Pears, "Adopting Competency Mindful of Professionalism in Baccalaureate Computing Curricula". EDSIGCON, 2019.

[22] P. Bourque and R. E. Fairley, "SWEBOK Guide V3.0", 2014. Available in: http://www.swebok.org.

[23] SBC - Sociedade Brasileira de Computação, "SBC Reference Curriculum for Undergraduate Courses in Bachelor of Computer Science and Computer Engineering". Grupo de trabalho responsável – CR2005, 2005.

[24] B. S. Bloom, "Taxonomy of Educational Objectives: The Classification of Educational Goals". Addison-Wesley Longman Ltd, 1969.

[25] A. P. Ferraz and R.V. Belhot, "Bloom's Taxonomy: theoretical review and presentation of the instrument's adequacies for defining instructional objectives". Gest. Prod., São Carlos, v. 17, n. 2, p. 421-431, 2010.

[26] K. Beck, "TDD - Test Driven Development". 1. Ed. Porto Alegre, Bookman Editora, 2010.

[27] I. S. Elgrably and S. R. B. Oliveira, "Gamification and Evaluation of the Use the Agile Tests in Software Quality Subjects: the Application of Experiments". 13th ENASE, 2018.

[28] ISO/IEC/IEEE 29119-1, "Concepts And Definitions". 2013.

[29] D. S. de Campos and D. J. Ferreira, "Plagiarism detection based on blinded logical test automation results and detection of textual similarity between source codes". 2020 IEEE Frontiers in Education Conference (FIE), Uppsala, Sweden, 2020, pp. 1-9, doi: 10.1109/FIE44824.2020.9274098.

[30] L. P. Scatalon, R. E. Garcia, and E. F. Barbosa, "Teaching Practices of Software Testing in Programming Education". 2020 IEEE Frontiers in Education Conference (FIE), Uppsala, Sweden, 2020, pp. 1-9, doi: 10.1109/FIE44824.2020.9274256.

[31] I. Santos, S. M. Melo, P. S. Lopes de Souza, and S. R. S. Souza, "Towards a unified catalog of attributes to guide industry in software testing technique selection.". 2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Porto, Portugal, 2020, pp. 398-407, doi: 10.1109/ICSTW50294.2020.00071.

[32] J. Grossmann, M. Felderer, J. Viehmann, and I. Schieferdecker, "A Taxonomy to Assess and Tailor Risk-Based Testing in Recent Testing Standards". in IEEE Software, vol. 37, no. 1, pp. 40-49, Jan.-Feb. 2020, doi: 10.1109/MS.2019.2915297.

[33] H. Aprahamian, D. R. Bish, and E. K. Bish, "Optimal Risk-Based Group Testing". Management Science 65(9):4365-4384. 2019. https://doi.org/10.1287/mnsc.2018.3138

[34] CMMI Institute, "CMMI Development v2.0", 2018, Accessed in 06/29/2018.

[35] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering (version 2.3)". Technical report, Keele University and University of Durham, 2007.

[36] G. Scallon, "Evaluation of Learning in a Competency Approach". Curitiba: Pucpress. 2015.

[37] L. N. Paschoal and S. R. S. Souza, "Planning and application of flipped classroom for teaching software testing". RENOTE - Revista Novas Tecnologias na Educação, 16, 2, 1–10. 2018.

[38] S. Laing and K. Greaves, "The Testing Manifesto". 2015. Available in: http://www.growingagile.co.za/2015/04/the-testing-manifesto.

[39] B. Marick, "Exploration Through Example: My Agile Testing Project". 2003. Available in Http://Www.Exampler.Com/Old-Blog/2003/08/21/ - Agile-Testing-Project-1.