A mapping of prescribed assets in the ACM / IEEE and SBC curriculum to the test design and execution practices of TMMi

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

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

***Abstract*— This Research to Practice Full Paper presents that the test design and execution activities are essential in a testing process, as they involve the identification and analysis of assets in order to develop and apply effective test cases in the discovery of incidents. In general, these activities are related to architecture, execution, and test analysis, where the TMMi (Test Maturity Model integration) prescribes a process area with many goals and practices to perform them in an organized and effective way. In the applicability of Exploratory Test (ET), most test design activities are generally not performed because many professionals understand that it is an agile test approach that does not have a systematic strategy that can be used, so only execution activities are performed based on the tester's intuition or experience. Additionally, it is observed in the specialized literature the growth in the use of ET in the industry, making it important to have a teaching-learning approach with structured activities allowing the understanding of how to apply test design and execution. Thus, this work aims to carry out a mapping to establish curricular assets to compose a syllabus aimed at teaching-learning activities of test design and execution. From this, have a basis to extract the curricular guidelines directed to the teaching-learning of ET. This is aligned with the research question: How to match assets related to knowledge on test design and execution of TMMi that adhere to the ACM/IEEE (Association for Computing Machinery / Institute of Electrical and Electronics Engineers) curriculum and the reference to training undergraduate course in Computing at SBC (Brazilian Computer Society)? The mapping took place sequentially: i) the study theme was defined, ii) the curriculum structures and their inputs to be analyzed were defined, iii) the description of each asset was identified, iv) the correspondence between the assets was made, and v) the mapping was validated using peer review. The activity "iv" was carried out based on the specific goals and practices related to the Test Design and Execution process area of TMMi to analyze the correlation of assets present in the SBC and ACM/IEEE curriculum guidelines. As a result, correspondence was generated on two levels, as follows: a) Training axes (SBC) and knowledge areas (ACM/IEEE) related to the test design and execution area of TMMi, b) Derived skills and contents (SBC), as well as topics and learning outcomes (ACM/IEEE) related to specific goals, specific practices and sub-practices of the TMMi process area. In general, the items listed above were identified, respectively, 13 assets and 110 items of assets adhering to the guidelines prescribed in the international and national curriculum for Test Design and Execution. The mapping encourages the development of a teaching plan focused on teaching and learning of ET design and execution, as there is a strong adherence of assets and their corresponding items with academic and industry practices. Such mapping should support the application of specific guidelines to ET.**

Keywords—curriculum, teaching and learning, test design and execution, software testing

I. Introduction

Several activities are necessary for a testing process with well-structured procedures aimed at achieving success. Such activities are related to the planning, design, execution, management, and evaluation of a test process to improve it and ensure that the quality of the product and / or software process reaches the highest possible level. For this, the tester can use many approaches (testing techniques and strategies) to detect defects [1, 2]. In this context, when the testing process is in line with guidelines prescribed in international and / or national standards or guidelines for good practice, it tends to be performed in a systematic way, achieving a significant level of efficiency and effectiveness in the discovery of defects, as these documents they are organized records of experiences in the market that combine theory and practice [3, 4].

Although there are international and national documents prescribing which procedures, goals, and practices can be established and performed in the testing processes from planning to evaluation, many professionals still do not use them as a basis for application in the processes in question [5]. One of the factors that caused this fact is that the market demands the delivery of products in the short term and with high quality, thus, there has been a growth in the use of agile test approaches [6].

In general, all areas related to the testing process are important, however, the activities related to planning, management, metrics, and measurement required by professionals with previous experience (sufficient maturity) so that they can easily identify the necessary resources, possible product risks and / or process, definition of optimal indicators that will allow measuring and performing process metrics and identifying the right actions to effectively manage the entire process according to the test approach used [7, 8]. In this way, Test Design and Execution activities become even more important in a testing process, as they are aimed at identifying and analyzing work products in order to develop and apply effective test cases in the discovery of incidents. Such activities are related to architecture, execution, and test analysis, where the TMMi (Test Maturity Model integration) prescribes a process area with several goals and practices to perform them in an organized and effective way.

The Test Design and Execution activities become even more necessary for learning in relation to other phases of the test process cycle, as it involves not only the specification and execution of tests but encompasses planning and control, analysis and design, evaluation of the exit criteria, and defect reports, as well as structured activities for closing defects [9, 10]. In this way, the great importance of teaching students how to carry out such activities in practice is notable, in order to ensure that the test process is organized, well defined, and effective, obtaining results in line with the test objective and what was previously designed [11].

In this circumstance, this present study aimed to investigate which assets of the guidelines contained in the Referential Training - RT for the Undergraduate Courses in Computing made available by the Brazilian Computer Society (in Brazilian portuguese, Sociedade Brasileira de Computação - SBC) and guidelines contained in the Computer Science (CS) Curricula made available by the Association for Machinery and Institute of Electrical and Electronics Engineer (CS-Curricula of the ACM/IEEE) relating them to the process area of Test Design and Execution could be useful to the knowledge of this area.

In the end, it allows obtaining an adherent mapping to the mentioned inputs, which should be the basis for detailed elaboration of teaching-learning plans for any test approach involving activities directed at Test Design and Execution. Such a teaching plan, therefore, should be aimed at undergraduate and graduate students in computer science in the perspective of providing adequate knowledge of how to carry out such activities with a well-structured and organized practical approach and not just limited to theory. Emphasize that, in this context, assets are structural elements that compose curricula or guides possible to be using.

In addition to this introductory section, the paper is structured as follows: Section II contains the theoretical foundation for a better understanding of the paper, section III contains the related works on this research, section IV describes the research methodology used in this paper, section V contains the mapping of assets, their description and justification, section VI contains the procedures used for mapping evaluation, section VII describes the expected results with this research, and section VIII describes the conclusions, limitations of this study and future work.

II. Theoretical Foundation

In this section, the concepts and description of reference curricula covered in this work are presented, more specifically about the Test Design and Execution process based on the TMMi, curriculum-related to ACM/IEEE, and the Training References for undergraduate courses in Computing at SBC.

A. Test Design and Execution Prescribed at TMMi

The TMMi structure for all process areas is made up of general and specific goals, practices, and sub practices, in addition, there are the work products for each practice. Among the many areas, the Test Design and Execution has the proposal to improve the capacity of the test process during the activities of architecture development (design), execution of tests and analysis from the establishment of specifications of architectural techniques, performing a structured test execution process, as well as managing incidents at closure [12].

In this context, structured testing implies the application of test design techniques with the possibility of using tools. These test design techniques are used to derive and select test cases and conditions from design requirements and specifications. A test case consists of the description of input values, preconditions, expected results, and postconditions. All this information is implemented as testing procedures, which are specific testing actions. Among such information, the specific test data required is essential to allow the execution of test procedures in an organized manner [12].

All these Test Design and Execution activities follow the test approach defined in the test plan. In this way, specific test design techniques (exploratory testing, black box, etc.) are based on the level and risks of the product identified in the planning. Finally, the test execution activities are summarized in the discovery, reporting, and evaluation of incidents directed to the closure. It is emphasized that all incidents found must be reported through the incident management system and communication to stakeholders must be carried out using established protocols [12].

B. Computer Science Curricula (CS-Curricula) of the ACM/IEEE

The IEEE and ACM computing society has made great efforts to establish international curriculum guidelines for undergraduate programs in computing in the last decades. Through the growth and diversification of the computing area, curriculum recommendations have also grown to include Computer Engineering, Information Systems, Information Technology and Software Engineering, in addition to involving Computer Science. These guidelines are updated regularly in order to maintain modern and relevant computer curricula. The last update launched on the Computer Science guidelines took place in 2013, having a body of knowledge redefined and highlighting the necessary foundations for a curriculum of that course. These updated guidelines also feature examples of courses and programs to provide more concrete guidance related to curriculum structure and the development of numerous institutional contexts [13].

In this way, they established principles for the Computer Science curriculum that ranges from the objectives expected for a student as to the level of organization of the institution. Regarding the context of skills expected from students, the principles define that such a curriculum should be designed to provide the graduates' ability to be flexible to act in different subjects, that is, it should prepare students for a variety of profession and, mainly, identify the skills and knowledge that students should have while providing greater flexibility in selecting topics. For this, three levels of knowledge description are established, which are organized into: Core Tier 1, Core Tier 2, and Elective [13].

For each course, the body of knowledge is presented as knowledge areas organized into topical themes instead of course limits. Each knowledge area is organized into units of knowledge that have topics according to the knowledge levels, also having the result of learning the topics in question. The topics described in Core Tier 1 should be a required part of the computer science curriculum with broad consensus for inclusion in all programs. The topics described in Core Tier 2 and Elective are understood to have limitations in the number of teachers available and other resources, so some of these topics may not be addressed. Regarding Core Tier 2, all topics are important, but they can be partially covered with at least 80%, while for Elective topics they are also considered very important, as a program serving only the basic material would provide insufficient breadth and depth in Computer Science. Many courses require some elective topics either by the established focus or are listed providing a basis for further study, and many courses can go beyond, even, the learning outcomes described [13].

C. Training References for Undergraduate Courses in Computing of the SBC

The SBC has been instrumental in the last few decades in relation to computer education in Brazil, as SBC has always brought up discussions of how undergraduate courses should be conducted. The SBC has participated in committees for the elaboration of Reference Curricula or discussed the ways of evaluating these courses in conjunction with the Ministry of Education of Brazil. In this context, it is emphasized that the National Curriculum Guidelines (in Brazilian portuguese, Diretrizes Curriculares Nacionais – DCNs) emerged from these curricula and discussions, which were approved in November 2016, through Resolution No. 05 of 11/16/2016 [14].

These discussions related to the teaching of computing at the undergraduate level take place in several events (congress, workshop, forum, symposium etc.) organized by SBC itself. From these discussions and preliminary studies, they elaborated the “Training References for Undergraduate Courses in Computing” (in Brazilian portuguese, Referenciais de Formação para os Cursos de Graduação em Computação – RF) for each of the courses contained in the DCNs: Computer Science, Computer Engineering, Software Engineering, Degree in Computing and Information Systems, including the technological graduation. It is noteworthy that the RFs are aligned with DCNs [14].

For each RF of the courses, there is a presentation, a brief history of the course, reference curricula, the benefits that the course offers to society, the aspects related to the professional training of the course, the profile of the graduate indicating competencies expected, the training axes, as well as the competences and contents that make up the RF for the course, the relationship of the competences described in the RF with the determinations of the DCNs, considerations on the performance of internships, complementary activities, and course completion works; the teaching and learning methodology, the legal requirements foreseen for the course, finally, thanks to several people who somehow contributed to the construction of the referred curriculum [14].

The methodology for preparing the RF adopts an approach guided by the competencies expected from the course's egress related to the contents involved in each competence. In this way, the RF was structured in order to understand that the expected profile of the graduate determines the general objective of the course, broken down into different training axes. The training axes aim to train graduates in generic skills. In order to reach each competence, several derived competences are listed, which determine the need to be developed in specific contents [14].

III. Related Works

Initially, there was a search in the specialized literature on asset mapping aimed at teaching and learning about Test Design and Execution and no work was identified focused on these activities. However, two similar works have been identified that conduct an asset mapping for broad teaching-learning of software testing and algorithms.

In this case, in [15] a mapping between different assets is presented in order to establish a knowledge base and be used as a reference for the development of a Teaching and Training approach to Software Testing with Agile Methods, as a whole. In this mapping, these authors used documents with guidelines for processes applied in the industry (TMMi and SWEBOK – Software Engineering Body of Knowledge) and academic or curricular (RF-SBC and CS-Curricula from ACM/IEEE) identifying more at the level of concept definition. If different from this, this present work did not use SWEBOK, because it is more directed to practical activities of Test Design and Execution, instead of trying to observe different concepts about the same term, technique, etc.

In the work of [16] it is also presented an asset mapping involving the SWEBOK documents and the ACM/IEEE RF-SBC and CS-Curricula curricula, but with a focus on the software process area with the purpose of enabling subsidy in the formation of course or software process subject based on industry and academia.

Therefore, there are some similarities with the other studies when observing the analysis inputs to carry out the mapping, with the objective of developing a teaching-learning approach. However, the differential of this work is to focus on the identification of assets that involve theoretical and practical activities of Test Design and Execution, which will support the elaboration of a teaching plan containing more practical subjects that are essential for the application of more effective tests in the industry.

IV. Research Methodology

This present work has been developed by performing the identification and cross-analysis of information in curricular guidelines or application guide referring to assets pertinent to Test Design and Execution activities. These analyzed documents are established by national and international reference organizations on the teaching and learning process, considering activities to be performed, content to be addressed, skills, objectives, and results to be achieved. In view of this, the following steps were taken:

- First, occurred the definition of the theme that would be addressed, where through a literature review it was possible to identify a great potential for study on education regarding the subject of Test Design and Execution,
- Subsequently, organizations and their necessary inputs to be analyzed were established, with this, it was identified: (i) TMMi has a guide describing activities in the area of Test Design and Execution, for example, describes the practices, goals, examples of work produtcs that can be involved in an organized way in the applicability of these activities in the industry; (ii) SBC, is the organization that contains curricular guidelines for computing specifically in the Brazilian context, then being used the training references for undergraduate courses in computing (RF-SBC); (ii) Association for Computing Machinery / Institute of Electrical and Electronics Engineers (ACM/IEEE), for also proposing issues related to computer education at an international

level, using the input called “Computer Science Curricula 2013” (CS-Curricula),

- Therefore, occurred an identification and description of each asset contained in the pre-established inputs previously. In this way, each one of these documents was analyzed, as well as, the structure adopted with the assets,
- Finally, the correspondence between the identified assets was analyzed. This activity was carried out based on the specific goals and practices described in the TMMi to then analyze the correlation of assets present in the curriculum guidelines. In this way, the reference among the assets in the adopted documents was obtained, generating an equivalence structure with the corresponding justifications for each relationship.

It is noteworthy that during the correspondence analysis there was a relationship between Learning Outcomes with Topics of the CS-Curricula from ACM/IEEE to correlate with the assets of the other adopted documents. In addition to having a full evaluation of this final mapping by an expert in Software Engineering.

V. Asset Mapping

The mapping of assets is a way of establishing correspondence with each other through many inputs, which deal with the same subject, which analyzed the assets related to the Test Design and Execution activities. For such crossing of information, the description, correspondence, and justification for the use of these assets will be presented.

A. Descripton of Assets

This section presents a description of all assets involved in the mapping. In this context, the assets used by TMMi were [3]:

- **Process Area** provides support and a more detailed specification of what is required to stabilize a defined verification and validation process,
- **Specific Goal** describes the unique characteristic that must be present to satisfy the process area. A specific goal is a required model component and is used in assessments to help determine whether a process area is satisfied,
- **Specific Practice** is a description of an activity that is considered important in achieving an associated specific goal. The specific practices describe the activities expected to result in the achievement of the specific goals of a process area. A specific goal is an expected model component,
- **Sub-practice** is a detailed description that provides guidance for interpreting and implementing a specific practice. Sub practices may be worded as if prescriptive but are an informative component meant only to provide ideas that may be useful for test process improvement.

The assets used in the SBC curriculum were:

- **Training Axis (F)** aims to train graduates in generic skills [14],
- **Generic or Derivative Competence (DC)** can express the knowledge, skills and attitudes expected from the course graduate, from the perspective of learning objectives, that is, it can be expressing the student's ability [17],
- **Contents (C)** are the set of values, knowledge, skills and attitudes that guide the graduate's competences [17].

The assets used for the ACM / IEEE curriculum were [13]:

- **Knowledge Areas (KA)** correspond to topical areas of study in computing. KAs are interconnected and that concepts in one KA may build upon or complement material from other KAs. Each KA is not intended to be in one-to-one correspondence with particular courses in a curriculum, but the curriculum will have courses that incorporate topics from multiple KAs which are structured by Knowledge Units,
- **Knowledge Units (KU)** represent individual thematic modules within a KA that are structured by topics,
- **Topics (Core Tier, Elective)** are structured in Core Tier 1, Core Tier 2 and Elective. Core Tier 1 are fundamental topics for the curriculum of any computing course. Core Tier 2 are generally essential topics where almost all or all of them must be addressed. Elective, are considered important topics in which a course must also give significant depth in many of the elective themes, not being restricted to core topics only.
- **Learning Outcomes (LO)** are a set of topics expected related to each topic specified. Each topic has a set of topics of learning outcomes students that must be achieved.

B. Correspondence Between Assets

From the analysis of the assets involved, as described in subsection A of section V, it was possible to generate the mapping of correspondence between each asset directed to the teaching-learning of Test Design and Execution. In Figure 1 it is possible to observe the direct or equivalence relationship by the equality of colors, for example, the process area, formation axis, knowledge area, and knowledge unit have correspondence. In general, it is highlighted that the assets without color (white background) were not correlated by the fact that the present study directs the analysis to specific practical approaches to Test Design and Execution from the TMMi, therefore, some assets did not obtain correspondence.

The corresponding assets among the inputs analyzed were mapped by similarity to their structural characteristics and prescribed items. In other words, the correlations between the assets were established considering “what” and “which” items are addressed, regarding the practices, expected results, contents, goals, and procedures always presented in the context of Test Design and Execution.

In view of this, the correspondence relationship between the Process Area (TMMi), Formation Axis (RF-SBC), and Knowledge Area and Unit (CS-Curricula) assets is exemplified, as they describe in general the issues involved or which items make up such an asset. In relation to the mapping established between the Specific Goals, Specific Practices, sub practices (TMMi), Derived Competences, Contents (RF-SBC), and Topics and Learning Outcomes (CS-Curricula) assets, as they describe in detail the objectives to be achieved and the practices, contents, and procedures that may be applicable in Test Design and Execution activities.

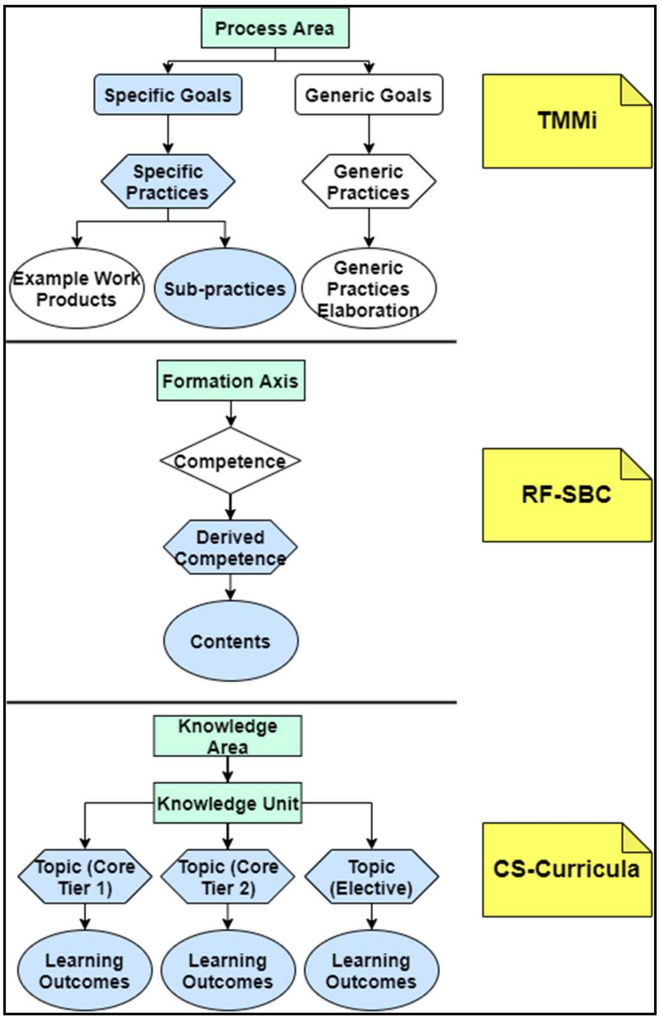

Fig. 1. Correspondence mapping of assets.

C. *Justification for Assets Mapping Between CS-Curricula, RF-SBC and, Test Design and Execution Prescribed at TMMi*

This section presents the correspondence of these assets in relation to the necessary knowledge about the Test Design and Execution process area, as well as a justification for this choice. The following equivalent assets were identified for this process area, which was organized by code to facilitate the visualization of the relationships between them.

For Formation Axis (F) are: (F1) System Development, (F2) System Deployment, (F3) Software Process and Management, (F4) Software Requirement, Analysis and Design, (F5) Software Construction and Testing, (F5) Software Quality. For Knowledge Area (KA) was selected Software Engineering with corresponding Knowledge Unit (KU): (KU1) Software Project Management, (KU2) Tools and Environment, (KU3) Requirement Engineering, (KU3) Software Design, (KU4) Software Verification and Validation, (KU5) Software Evolution.

Due to space limitations in the paper, a list of all selected asset items was made available at the link *https://bit.ly/37Hzzd6*. In this list, it is possible to observe the code identifier for each item ("List of Assets" tab), and thus understanding in detail the correspondence relationship of the assets presented in Figure 1 ("Mapping" tab). It is noteworthy that for understanding the second level of equivalence and consequently of the data exposed in Tables I, II, III and IV, the codes of the items of the assets are: Specific Goal (SG), Specific Practices (SP), Sub-practices (SBP), Derived Competence (DC), Contents (C), Topic (T), Learning Outcomes (LO). For example, the code "F1F2DC1" means that the first competence derived from the list (DC1) is related to the first two training axes (F1 and F2), thus, it is possible to understand the other codes.

The Specific Goals (SG) of TMMi will guide the presentation of the mapping below on the Test Design and Execution process area.

In SG1 - Perform Test Analysis and Design using Test Design Techniques, such asset items (as can be seen in Table I) are related by presenting the proposal to design the test cases, both for static and dynamic test approaches. Initially, to guarantee quality criteria throughout the development stages, it is essential that all necessary resources are presented and analyzed to derive the test conditions (static and/or dynamic), define the test limitations according to the domain of the object under test. In this case, one should consider both the documentation (work products) available and test design strategies, as well as the tools that can be used, in addition to needing to clearly describe the problems that can be addressed by these tools.

TABLE I. ASSET ITEMS RELATED TO THE SG1 AND ITS SPECIFIC PRACTICES

ID	Assets					
SG1	Perform Test Analysis and Design using Test Design Techniques	Sub-practices	Topic	Learning Outcomes	Derived Competence	Contents
SP 1.1	Identify and prioritize test conditions	SP1.1SBP1	KU2T3, KU2T4, KU4T9, KU5T11, KU5T12, KU5T13, KU5T14, KU5T15	KU2LO3, KU2LO4, KU4LO10, KU5LO12, KU5LO13, KU5LO14, KU5LO15, KU5LO16, KU5LO17, KU5LO18, KU5LO19	**F1F2DC1,** F4DC4, F4DC5	C1, C2, C3, C4
SP 1.2	Identify and prioritize test cases	SP1.2SBP2	KU5T13	KU5LO16	F4DC4, F4DC5, F5DC8	C2, C3, C4
SP 1.3	Identify necessary specific test data	SP1.3SBP3				
SP 1.4	Maintain horizontal traceability with requirements	SP1.4SBP4	KU3T7, KU4T8, KU4T10	KU3LO8, KU4LO9, KU4LO11	F4DC4, F4DC6, F4DC7	C5, C6, C7

From this, test cases or charters for exploratory testing are generated from techniques of static and dynamic verification and analysis of software work products in line with pre-defined test conditions. In addition to identifying the test cases, it is possible to establish at the same time which specific test data is necessary to carry out the tests. In this context, the traceability of requirements is perceived by the relationship with the test cases, where it is possible to identify the coverage of test cases in relation to the requirements. In this case, all the necessary changes from the correction of defects become traceable and understandable. Such activities are part of requirements management and are directly related to testing activities.

In this way, it is possible to design test cases or adherent charters with the functionality intended for the test submission, following the defined priorities. Therefore, the test strategy is identified according to the available documentation, defined objectives and the customer's need, considering the tools (support resources) to perform such testing activities.

In addition to this ordered list of activities that follows the specific practices prescribed in the TMMi, it is possible to highlight a strong correspondence between the items of assets in other documents analyzed. Thus, it is exemplified that to achieve the specific goal "Perform Analysis and Test Design Using Test Design Techniques" it is necessary to perform some specific practices (for example, "Identification and Prioritization of Test Conditions" - SP1.1), which guide the organized execution of a set of subpractices (for example, test basis analysis, selection of appropriate test design techniques, derive and prioritize test conditions - SP1.1SBP1).

Correlated to this, it is mentioned the topics Requirements analysis and design modeling tool (KU2T3) and Static and / or dynamic test analysis tools (KU2T4) that are adherent to some expected learning results from students Discuss the limitations of test (KU5LO19), Describe techniques for identifying significant test cases (KU5LO15), Describe and distinguish between different types of tests (KU5LO14), Describe how to make tools available for static and dynamic testing (KU2LO3). In line with this, some derivative skills and contents are also mentioned, for example, applying techniques for software solution design (F4DC5) and software design (C3), respectively. Thus, it is possible to observe in the spreadsheet the organization of correlation to other items of assets.

In SG2 - Perform Test Implementation, the assets are related (as can be seen in Table II) because just as test case identification techniques are applied, a strategy for implementing test cases must be applied in an organized way, being properly documented and reviewed. For any identified and implemented test case, it is also necessary to elaborate the specific test data necessary to execute the test procedures properly. All these definitions must be documented for eventual assessments of correctness and compliance with the requirements. In this way, maintain the relationship between the assets using techniques for static and dynamic verification and analysis of software work products. Regarding the activity itself, creating an execution schedule, are related to the project management activity, where it defines such a work schedule allowing the tracking of test activities.

TABLE II. ASSET ITEMS RELATED TO THE SG2 AND ITS SPECIFIC PRACTICES

ID	Assets					
SG2	Perform Test Implementation	Sub-practices	Topic	Learning Outcomes	Derived Competence	Contents
SP 2.1	Develop and prioritize test procedures	SP2.1SBP5	KU5T13	KU5LO16	F4DC4, F4DC5, F5DC8	C2, C3, C4
SP 2.2	Create specific test data	SP2.2SBP6				
SP 2.3	Specify intake test procedure	SP2.3SBP7	KU5T13	KU5LO16	F4DC4, F4DC5, F5DC8, **F5F6DC9**	C2, C3, C4, C8, C9
SP 2.4	Develop test execution schedule	SP2.4SBP8	KU1T1, KU1T2, KU5T17	KU1LO1, KU1LO2, KU5LO20, KU5LO21, KU5LO22	F3DC3, **F5F6DC9**	C8, C9, C10,

To exemplify a more detailed horizontal correlation of SG2 with the items of assets present in the analyzed documents, is mentioned the need to develop and prioritize test procedures (SP2.1), as well as to create specific test data (SP2.2) that it consists of carrying out the subpractices of elaboration the procedures in an organized way and in line with the test cases and test conditions (SP2.1SBP5), in addition to defining data to allow the execution of the tests (SP2.2SBP6). In this context, it is noted that the topic on generating test cases (KU5T13) is strictly directed to the learning result in knowing how to create and document a set of tests (KU5LO16). Adhering to this is the derived competence that involves the implementation and documentation of software solutions (F4DC5) and the content of software design (C3). In this way, it is possible to observe the correlation for the other items of assets in relation to SG2.

In SG3 - Perform Test Execution, the assets are related (see Table III) as the TMMi prescribes the performance of initial tests in order to verify the basic functionalities to observe whether the object under test can receive all the planned tests. From this, all test procedures are carried out as planned (identified and implemented) to detect incidents, as well as having to record them in a tool that can be managed, from detection to correction. All this also includes testing for non-code work products. It is noteworthy that only the assets of the ACM/IEEE CS-curriculum are clearly related to the activity of reporting incidents. Such assets of ACM/IEEE are aligned both in the use of a tool to report incidents and to carry out the review of these reports (work products). In contrast, in the RF-SBC there is no topic that makes clear any direct relationship with incident reporting, but there are activities related to the detection of incidents that can be more broadly linked.

TABLE III. ASSET ITEMS RELATED TO THE SG3 AND ITS SPECIFIC PRACTICES

ID	Assets					
SG3	Perform Test Execution	Sub-practices	Topic	Learning Outcomes	Derived Competence	Contents
SP 3.1	Perform intake test	SP3.1SBP9	KU5T12, KU5T14, KU5T16, KU5T18, KU5T19, KU5T20, KU5T21, KU5T22	KU5LO18, KU5LO23, KU5LO24	F5F6DC9, F6DC10	C2, C4, C11
SP 3.2	Execute test cases	SP3.2SBP10	KU5T12, KU5T16, KU5T18, KU5T19, KU5T20, KU5T22	KU5LO23, KU5LO24	F5F6DC9, F6DC10	C2, C4, C11
SP 3.3	Report test incidents	SP3.3SBP11	KU5T14, KU5T18, KU5T21	KU5LO18, KU5LO24	**F5F6DC9**	C8, C9
SP 3.4	Write test log	SP3.4SBP12				

To exemplify the achievement of the specific goal of "Perform Test Execution", the specific practice of executing test cases (SP3.2) is mentioned, where it is necessary to execute it following the documented procedures comparing the expected result with the current result (SP3.2SBP10). Correlated to this, it exemplifies with the topic of applying static and/or dynamic test approaches (KU5T18) involving many types of tests (KU5T12), including review or inspections where it is possible to observe that the learning results are adherent, waiting that the student can conduct inspections or tests, as part of a team activity, in a code segment (KU5LO23 and KU5LO24). In this context, there is an alignment of the derived skills on preventive defect detection (F6DC10) and application of verification and validation techniques and procedures (F5F6DC9), as well as the contents of software testing (C11) and Verification and Validation (C2) are strongly correlated.

In SG4 - Manage Test Incidents to Closure, the assets are related in general (as can be seen in Table IV) since the activities of analysis of the incidents (defects) directed to the closure are clearly related to the process of maintenance and evolution of the software. In this case, the activities are focused on managing the changes that occurred by correcting an incident, performing a regression test or retesting to validate certain functionalities affected by the incidents.

TABLE IV. ASSET ITEMS RELATED TO THE SG4 AND ITS SPECIFIC PRACTICES

ID	Assets					
SG4	Manage Test Incidents to Closure	Sub-practices	Topic	Learning Outcomes	Derived Competence	Contents
SP 4.1	Decide disposition of test incidents in configuration control board	SP4.1SBP13	KU2T5, KU2T6, KU5T13, KU6T23, KU6T24, KU6T25	KU2LO5, KU2LO6, KU2LO7, KU5LO17, KU5LO18, KU5LO19, KU6LO25, KU6LO26, KU6LO27, KU6LO28, KU6LO29, KU6LO30	F3DC2	C7, C12
SP 4.2	Perform appropriate action to fix the test incident	SP4.2SBP14				
SP 4.3	Track the status of test incidents	SP4.3SBP15				

In line with these activities, the topics involving release management and version control (KU2T5 and KU2T6) are exemplified, as well as the execution of regression test (KU5T13) being expected as a result of learning that the student will be familiar with how version control can be used to manage software releases (KU2LO6), how to apply regression testing (KU2LO17), defect tracking (KU2LO18) and identify problems related to software evolution (KU2LO25). In this context, it is exemplified that the derived competence on application of techniques and procedures for maintenance and software development (F3DC2), as well as the content involving maintenance, refactoring, debugging and impact analysis (C12) identified are strongly adherent.

In summary, it is obvious that the importance of each asset mapped in the analyzed documents is highlighted. Where this correspondence of assets involves necessary and appropriate subjects (activities, concepts, procedures, etc.) being strongly aligned with each other in order to provide that a teaching plan can be developed involving content that can provide the student with sufficient skills to apply in practice effectively and efficiently the activities directed to Test Design and Execution.

VI. MAPPING EVALUATION

After executing the mapping, the document was submitted to the evaluation of an expert in Software Engineering to validate it. Such an evaluation allowed to carry out an adequate refinement, being able to identify a set of assets and items of those assets strongly adherent to the analyzed inputs.

A. Evaluator Profile

The evaluator is a professional in Software Engineering, being certified as a consultant-implementer and leader appraisal of the MPS.BR - Model for Improvement the Brazilian Software Process, CMMI – Capability Maturity Model Integration, CERTICS, MEDE-PROS models, among others, having already implemented and evaluated these models in more than 40 enterprises in different regions of Brazil. In addition, he already has numerous papers published internally on Software Engineering education, obtaining awards for the prominence in some developed works.

B. Mapping Refinement

The refinement was a process carried out by peer review techinique where the assets initially identified were analyzed to make sure that the relationship established was coherent. Subsequently, the asset items extracted as necessary were analyzed to ensure that an adequate and adherent mapping to the curricula used was obtained.

In the revision of the assets, there was only a refinement in the relationship of the assets, which occurred with the insertion of the asset Learning Outcomes (ACM/IEEE). This adjustment was necessary, as it was perceived that it was strongly related to subpractices (TMMi) and content (RF-SBC). On the other hand, in the review of asset items, the need for further adjustments involving the inclusion or insertion of Knowledge Unit, topic and Learning Outcomes for CS-curriculum of ACM/IEEE and Training Axes, Derived Competencies and Contents for RF-SBC was perceived (as can be seen in Figure 2).

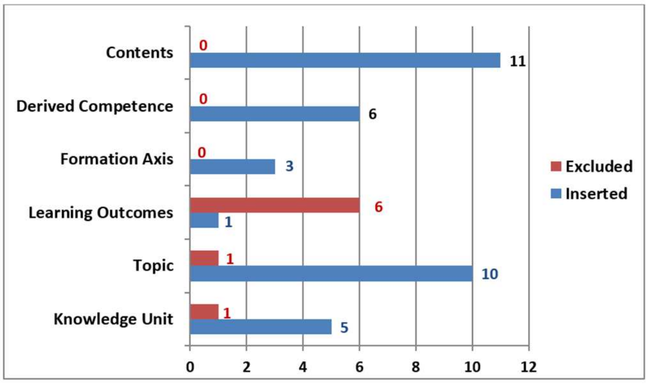

Fig. 2. Asset items adjusted in the peer review.

Regarding the inclusion of asset items, it is mentioned that topics that involve the relationship and traceability between requirements and design are correlated with the specific practice that establishes the maintenance of screening between work products (SP1.4). In addition, topics were included that involve the elaboration of a test execution schedule (SP2.4) and derived competences related to the application of static and/or dynamic verification and validation techniques and procedures (F5F6DC9), maintenance and evolution software (F3DC2), as well as considering content on static and/or dynamic review techniques and analysis of software work products (C8 and C9).

In relation to the assets that were disregarded from the mapping because it is not directly considering the context of Test Design and Execution, the topic on failure estimation approaches (incidents) and the knowledge unit on development methods are cited. Although this knowledge unit basically involves procedures for conducting code review, however, the knowledge area of Fundamentals of Software Development and not directed to Software Engineering is inherent, which was the area of study in focus of this work.

VII. EXPECTED RESULTS

Initially, the mapping allows an aligned analysis of the Test Design and Execution activities with the main reference documents (curricula) on teaching and knowledge in Software Engineering. In view of the mapping carried out it is possible to identify and understand the correlation between the assets and their items relevant to the knowledge of Test Design and Execution at an international and national level. Thus, the eligible guidelines are strongly aligned with academic and industry practices.

It is noteworthy that there were 13 assets and 110 items of assets involved in a correspondence on two levels, being: a) Training axes (RF-SBC) and knowledge areas (ACM/IEEE) related to the Test Design and Execution Project process area of TMMi, b) Derived competences and contents (RF-SBC), as well as the topics and learning outcomes (ACM/IEEE) relate to the specific goals, specific practices and sub-practices of the TMMi process area in question. With this, the mapping makes it possible to observe about the main competences expected by the egress in undergraduate courses in computing defined at national and international level, regarding the approach of the Test Design and Execution process area.

Therefore, it is expected that from this mapping it is possible to have sufficient guidelines to support the elaboration of curricula (teaching plans) specific to the teaching and learning of Test Design and Execution for any desired test approach.

VIII. Conclusion And Future Work

This work presents a mapping of assets and their corresponding items prescribed in national and international curriculum guidelines. The mapping encourages the elaboration of a curriculum focused on teaching-learning of Test Design and Execution, as there is a strong adherence of assets and their corresponding items with academic and industrial practices. In view of the mapping, it is also possible to observe the students' knowledge and skills about the test area in question.

Therefore, this work provides with the mapping carried out a reference on the main contents and competences necessary to develop the teaching and learning process of Test Design and Execution, which can be used as a basis to develop a teaching plan aimed at any testing approach specifically involving such activities. In other words, mapping becomes an important tool for future collaboration in developing new teaching approaches in this test area, as well as selecting which more specific guidelines are useful for certain test approaches.

As future work it is expected to develop a curriculum with the identified assets for teaching-learning of Test Design and Execution and, from that, to establish specific curricular guidelines to develop a student-centered teaching-learning approach by applying Exploratory Test Design and Execution activities.

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] G. J. Myers, "The Art of Software Testing". Second Edition. John Wiley & Sons, Inc.2004.

[2] M. Roper, "Using Machine Learning to Classify Test Outcomes". IEEE International Conference on Artificial Intelligence Testing (AITest). DOI 10.1109/AITest.2019.00009. 2019.

[3] A. N. Crespo, O. J. da Silva, C. A. Borges, C. F. Salviano, M. T. A. Junior, and M. Jino, "A Methodology for Software Testing in the Context of Process Improvement". Simpósio Brasileiro de Qualidade de Software - SBQS. 2004.

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

[5] I. E. F. Costa and S. R. B. Oliveira, "An Evidence-Based Study on Automated Exploratory Testing". 17th International Conference On Information Systems & Technology Management – CONTECSI – 2020.

[6] M. Cohn, "Succeeding with agile: software development using Scrum". Pearson Education. Addison-Wesley Professional; 1 edition. 2009.

[7] F. Jorge, C. Bezerra, E. Coutinho, J. Monteiro, and R. Andrade, "The Evolution of the iTestLearning Game for Teaching Software Testing Execution Activities". Nuevas Ideas en Informática Educativa – TISE. 2015.

[8] V. Farias, C. Moreira, E. Coutinho, and I. Santos, "iTest Learning: A Game for Teaching Software Test Planning". V Fórum de Educação em Engenharia de Software (FEES 2012). Natal – RN. 2012.

[9] R. Jain and R. Kaluri, "Design of Automation Scripts Execution Application For Selenium Webdriver and Testng Framework". ARPN Journal of Engineering and Applied Sciences. ISSN 1819-6608. VOL. 10, NO. 6, APRIL. 2015.

[10] T. Müller, D. Graham, D. Friedenberg, and E. Veendendal, "Knowledge Base for Test Certification - Foundation Level Syllabus". ISTQB - Comissão Internacional para Qualificação de Teste de Software. 2007.

[11] L. Copeland, "A Practitioner's Guide to Software Test Design". Artech House Publishers. ISBN: 1-58053-791-X. 2004.

[12] E. Van Veenendaal, "Test Maturity Model integration – TMMi: Guidelines for Test Process Improvement". Produced by TMMi Foundation. 2018.

[13] ACM/IEEE, "Computer Science Curricula 2013". ACM and IEEE Computer Society, Incorporated: New York, NY, USA. 2013.

[14] Sociedade Brasileira de Computação – SBC, "References of Training for Undergraduate Computing Courses". Retrieved May, 11, 2018. 2017.

[15] I. S. Elgrably and S. R. B. Oliveira, "A Proposal for Teaching and Applying Focused of Agile Tests Methods made by an Assets Mapping". 15th International Conference On Information Systems & Technology Management – CONTECSI – 2018.

[16] J. A. S. Quaresma and S. R. B. Oliveira, "A mapping of the assets included in the reference standards of RF-SBC, CS-Curricula ACM / IEE 2013 and SWEBOK regarding the knowledge area about Software Process". 17th International Conference On Information Systems & Technology Management – CONTECSI – 2020.

[17] M. Van der Klink, J. Boon, and K. Schusmans, "Competences and higher professional education: present and future". Revista Europeia de Formação Profissional, 40(1), 72-89. 2007.