# A Teaching Method for Software Measurement Process based on Gamification

Lennon Sales Furtado, Sandro Ronaldo Bezerra Oliveira
Graduate Program in Computer Science
Federal University of Pará
Belém, Pará, Brazil
e-mail: lennonsfurtado@gmail.com, srbo@ufpa.br

***Abstract*—The value of an effective measurement program lies in the ability to control and predict what can be measured. Thus, the measurement program has the capacity to provide a basis in decision-making to support the interests of an organization. This means it is only possible to run an effective measurement program with a team of software engineers who are well trained in this area. However, as the literature shows, there are few computer science courses that include the teaching of software process measurement in their program. Even these, generally only discuss the basic theoretical concepts of this process with little or no measurement in practice, which discourages the students from's learning the measurement process. In this context, according to some experts in software process improvement, one of the most widely used approaches to maintaining the motivation and commitment to improving the program, is the use of gamification. In light of this, the aim of this paper is to set out a proposal for teaching the gamification measurement process. This seeks to improve student motivation and performance in carrying out tasks related to software measurement, by incorporating elements from games into the measurement process, and thus making it more attractive for learning.**

***Keywords-education; gamification; teaching method; software engineering; software process measurement.***

## I. INTRODUCTION

The purpose of the software measurement process is to collect, store, analyze and report data on developed products and implemented processes within an organization, in order to support organizational goals [1]. Moreover, the importance of an efficient measurement program lies in its ability to control what can be measured [2]. By being able to control the metrics of the measurement program, the organization will be capable of predicting organizational and marketing behavior [3].

Even though the measurement process is of such importance to Software Process Improvement (SPI), the software industry has been reluctant to employ efficient measurement programs [4][5]. This is because many software managers and software engineers, including academics in software engineering and computer science, seem to have little or no practical knowledge of this subject [6].

In general, in every software measurement program, what determines its success is the human factor, because if there is a lack of commitment to this program, it is unlikely to achieve the desired results, i.e., the visibility and control of software metrics to aid in decision-making. In this context, one of the most widely used methods to maintain the motivation and commitment of the people involved in a SPI program is the use of gamification [7].

Gamification by definition is the use of game elements and game design techniques outside their usual context [8]. This process seeks to improve the commitment, motivation and performance of a user when carrying out any task, by incorporating features of games and game mechanics, and thus, make the task more attractive [9].

By means of this educational tool, this paper seeks to address the problem of teaching the measurement process by exploring aspects of gamification. This involves adopting and evaluating an approach for the use of this tool as a motivating factor in the teaching of software process measurement.

In addition to this introductory section, this paper is divided into the following sections: Section II will cover factors that explain and identify the question under study. Section III will outline the problem addressed in this paper and discuss related work and the limitations of its approaches. In Section IV, a number of research questions will be raised that will guide future investigations covered by this research. In Section V the methods used to answer the research questions will be examined in detail, and finally, in Section VI the expected results will be discussed, together with a report on the progress of the research.

## II. A GAP IN THE AREA AND SCOPE OF THE RESEARCH

This section describes the gap in the area and the scope of the research discussed in this paper.

### *A. A Gap in the Area*

As pointed out by Jones [6], many software managers and software engineering professionals are unaware of the key aspects of measurement (planning, preparing, collecting data and analyzing them for decision-making). One factor in this problem that must be taken into account is the way the process of measurement and analysis is taught.

This concern is directly related to the fact that the measurement process is generally regarded as difficult and time-consuming [8][10][11][12]. An initial assumption that must be made when seeking to understand this problem, lies in the way this discipline is taught [13], since it has been neglected in the undergraduate curriculum and its importance is not stressed enough to encourage students to learn in practice [14]. In addition, another serious factor is the lack of guidelines on how to implement the practice of measurement [15][16][17].

In other words, the dynamics of expository lectures,

 ISBN: 978-1-61208-590-6 

(where students passively construct their knowledge), tend to be very time-consuming and have inherent weaknesses with regard to the difficulties of transferring knowledge to real life situations [18].

Furthermore, the measurement process is only one of several processes taught in the software quality assurance disciplines, and because of this, there is little time to show its practical application. This is the main obstacle to the establishment of knowledge in the measurement process. It means that the students clearly have difficulties in applying software measurement to real-life situations.

In attempting to meet the needs of the students involved in Software Quality Assurance, this research will set out a proposal for the gamification of the software measurement process. This is a way of overcoming the problem of a lack of opportunity to practice this process in the undergraduate curriculum, by introducing the concepts of basic measures, derived measures, indicators, Goal Question Metrics (GQM) [19] and a Practical Software Measurement (PSM) program [20].

### B. Scope of the Research

This study will investigate the following: the conceptual factors involved in the development of the proposal, and the implementation and evaluation of the gamification tool for teaching the software measurement process. In addition, it should be mentioned that this research has the following objectives:

- To set out a teaching proposal for the software measurement process by the application of gamification,
- To analyze the state-of-the-art in the use of gamification for the teaching of software processes,
- To identify different approaches to the use of gamification, which can provide the user with greater ease when learning the discipline of software measurement,
- To identify the limitations of gamification as a teaching method,
- To examine the concepts of basic and derived units of measurement,
- To define the concept of indicators,
- To set out the GQM and PSM paradigms,
- To predict the expected outcomes and practices contained in the Nationwide Program for Software Processs Improvement in Brazil (MPS.BR-SW) and Capability Maturity Model Integration (CMMI-DEV) programs, which include the measurement processes,
- To describe the most widely used metrics in the market, such as metrics for product maintenance, performance, and reliability,
- To describe the metrics that do not depend on the programming language,
- To define the concepts of the Organizational Measurement Plan, GQM Plan and Measurement Report,
- To evaluate the teaching proposal by comparing it with traditional teaching methods

Apart from these objectives, this research aims to prove or refute the following hypothesis: the gamification research proposal to teach software process measurement is an appropriate way of motivating students to acquire the necessary skills for its practical application.

## III. PROBLEM STATEMENT AND RELATED WORKS

The research problems can be categorized into three groups, which are as follows:

- The need to analyze the measurement practices used in the software industry, i.e., to determine which activities are really useful for the training of a software engineering professional,
- The need to analyze the references for a curriculum and the teaching approaches adopted by teachers in the area, to identify which measurement practices are covered,
- To identify the different approaches in the use of gamification, which provide the user with greater ease when learning the software measurement discipline.

### A. The Background of the Teaching of Software Measurement

According to Bass [21], software measurement can be defined as a quantitative assessment of any aspect of software processes and products. It allows a better understanding of these areas and thus helps in planning, controlling and improving what is produced and how it is produced.

In summary, the measurement program is designed to generate information on products, processes and people. This kind of information serves as a framework for decision-making, which can guide organizations and their projects [22]. That is, it is a very important process for organizations and for programs designed for software process improvement.

However, organizations often make complaints about students who enter the job market, usually on the grounds that they are not prepared to tackle the real problems found in industry [23]. This is due to the difference between the industrial environment and academic programs [24]. For a better understanding of this problem, it is necessary to check how software measurement is being addressed in academic courses of computer science, and if organizations think that students who enter the job market have a sufficient knowledge of software measurement. These questions were raised and explored in a study carried out by [25] and its results are summarized below:

- A survey answered by students and teachers pointed out that the software engineering course is generally mandatory in graduate programs, while software measurement is, in most cases, an optional course,
- All the teachers and students who took part in the survey indicated that the teaching of software measurement is mainly given in expository lectures

 ISBN: 978-1-61208-590-6 

and more than 50% through a case study. In addition, in some cases, students learn from another approach, such as applying measurement strategies in real projects,

- Software measurement courses are usually taught in graduate programs, although these courses are mostly optional,
- The level of student learning is usually assessed by written exams (75%) and by projects (58%).

To meet the needs of industry, it is necessary to prepare students for these environments and their real problems. There are many approaches for this, including in-context learning, where the student learns to use knowledge in a context with the same real-world challenges [26]. That is, the student will learn software process measurement by measuring actual software, as well as by designing measurement plans and putting them into effect.

Another approach is the application of teaching by Problem Based Learning (PBL), which follows the principle of learning by solving problems or addressing challenges related to the practice of software measurement.

In addition to these approaches, one of the approaches found in the literature is the use of serious games for teaching a subject [18], where the student plays a game as an educational tool to introduce the theoretical concepts and simulate the practical application of these concepts As a result, it can be seen that an important feature in the teaching of software measurement, is to adopt innovative approaches in the way the education will be conducted [17]. This is because universities have a myriad of student profiles with different levels of interest and motivating factors that will lead them to obtain the desired knowledge from an educational institution.

Among these new approaches is the use of game elements in terms of mechanics and dynamics, which are a motivational factor in teaching or carrying out a task. This approach is known as gamification and seeks to improve the commitment, motivation and performance of a user when learning a subject or carrying out a task [9]. In addition, one of its great advantages is the familiarity of the students of this generation with games, because they have grown up with them and actively play games as both a form of entertainment and learning.

In summary, the objective of this research is the development and validation of an educational tool that uses gamification for the teaching of software measurement.

### B. Problem Areas

Despite its importance in industry, in many cases, the measurement process has failed to yield benefits to organizations.

Following a survey conducted in Brazil in 2012 by the American Chamber of Commerce (AMCHAM) [27] with 44 Information Technology (IT) executives, it was found that 86% of the executives interviewed were not satisfied with the way the measurement was conducted in their organizations. Among the main difficulties highlighted were the following: a) obtaining tangible benefits and producing a return on measurement activities (41%), b) establishing performance indicators (30%), c) obtaining information on the impact of IT on other sectors of the company (18%), and d) quantifying the efficiency of the processes and systems (14%).

All the difficulties found arise from the capacity of the professionals responsible for the measurement process, since they fail to conduct a process efficiently that covers all aspects of measurement (measuring, storage, analysis and reporting).

The problems reported by the industry are just a few of the symptoms of an aging education in software measurement. This is corroborated in Jones's paper [28], which found that there were 28 problems that need to be addressed while measuring software, the most serious being the absence of a proper training system for students, to enable them to enter the world of industry with real problem-solving skills.

In addition to the problems pointed out by Jones [28], the literature states that software measurement is a complex and time-consuming task [8][10][11][12]. Apart from these problems, software measurement in education faces other challenges, such as being one issue among the 83 topics covered by software engineering and the fact that in most courses, it is treated as an optional subject [25]. Moreover, there are the problems arising from the dynamics of expository lectures, where students passively construct their knowledge, and thus tend to waste a lot of time. There are also the inherent weaknesses in this methodology with regard to the difficulties in the transfer of knowledge to real-life situations [18]. This makes it difficult for the student to understand the subject in depth, since this is only possible when a student is motivated and engaged with the subject. However, it can be achieved by exposing these students to situations that allow them to participate in problem-solving activities and tasks related to the issue of software measurement. Hence, these difficulties provide opportunities for improvement, and will thus lead to the maturity of the software measurement process. However, this process is still regarded as an “immature” field [23], due to the lack of a consensus on international software standards for measurement [29] and divergences in the implementation and interpretation of software metrics with tools [30]. That is, it is a field of study that needs to undergo several improvements and also be standardized.

### C. Limitations of Related Works

With regard to related works, only those will be evaluated that provide mechanisms for teaching software engineering through gamification or serious games.

The closest approach to this research is in the exploratory study conducted by Gresse von Wangenheim [18], which employs a serious game (X-MED) for teaching software measurement. In this work, the student plays the role of a measurement specialist who has to carry out the measurement process in a movie rental company. As the users progress in the game, they must answer some questions and earn points, so that they can produce a result that corresponds to the level of learning of the student.

The most significant contributions made by this system

are:

- It provides a comprehensive study that covers all the stages of creating the proposed game,
- It provides a complex theme that can be measured by a game that simulates a situation which involves a real application of this process,
- It validates the game proposed by applying it in undergraduate classes so that it is able to evaluate the degree of acceptance and benefits of the game in practice by pre and post questionnaires with the users.

However, the game devised by Wangenheim did not show any improvements in teaching measurement when compared with the traditional teaching method based on lectures. This can be attributed to some weaknesses in this work, such as:

- The game makes use of game mechanics and elements that are not very attractive, for example mechanics quizzes. The game mechanics and their related elements are determining factors to motivate and engage the student,
- The aesthetic appeal and sound of the game are not attractive; this is a very important factor in maintaining the user's interest in the game,
- The game is not suitable for mobile devices. This is a weakness, as most students are used to making use of this platform as a means of entertainment. Thus, it is more likely that students will make use of the game in their free time.

The works [31][32][33] were also evaluated, which made use of gamification or serious games for the teaching of software engineering.

In the work conducted by Bartel [31], which includes a gamified course for the teaching of design patterns, the students were encouraged to work as a team. Different tasks were assigned to each of the students in the team and each team had to find solutions to the problems raised in the classroom. On the completion of every task, the students were awarded a score by their classmates and the teacher. In this work, the following difficulties were detected:

- The proposal is very simple and limited since it is basically a quest list,
- Feedback is given by classmates and not by an automated system, which meant it was based on the subjective opinions of the students,
- The evaluation of the results of the game was not compared with the traditional teaching methods to validate its usefulness.

In the course of the paper [32], which employed a gamified classroom for teaching extreme programming, the students who took part, stated in the questionnaires that they thought the learning experience was good when compared with the learning experience that involved conventional lectures.

The participants showed an improvement in learning and coding performance after they had become used to the gamification teaching method. Despite this there were a number of limitations in this paper which are listed below:

- The lack of immediate feedback and transparency in the data collected, as students were only given an assessment of their progress at the end of the cycle,
- The experiment undertaken in the paper needs a special kind of class where students work 8 hours a day and use gamification as a part of the teaching method,
- In the paper, few topics are discussed about the planning of the gamification (e.g. the game elements and mechanics), only the results of the experiment are given.

In addition, the study conducted by Chaves [33] included a serious game, which taught how to design software processes, and made evaluations of the pre and post questionnaires that were employed in undergraduate classes on computer science. The class that took part made a significant contribution to the efficacy of learning and the application of acquired knowledge in its results. Nonetheless, the study had the following drawbacks :

- The constraints imposed by the game can restrict the creativity of the player, because only the traced path can be followed,
- The students only memorize the proposed models rather than learn them,
- There are few levels in the game and hence a considerable increase in difficulty, which distorts it.

## IV. QUESTIONS, HYPOTHESES AND DISCUSSION

The main goal of our research is to set out a teaching method based on gamification for software measurement, as an educational tool for computer courses, and this approach is based on the results of different kinds of research methods such as: a survey, a systematic review of the literature, a literature review of the the curriculum guidelines and user testing.

In addition, we found many references [6][13][14][21] that support our initiative and point to the need to approach software measurement education in a non-traditional way.

The following research questions should be addressed to understand the needs of students and industry, as well as the accuracy of the gamification system :

- **RQ1.** What is the state-of-the-art on software measurement education when gamification is used as an educational tool?
- **RQ2.** How can educators benefit from gamification in measurement education and learning?
- **RQ3.** What are the metrics and indicators that are most widely used by the software industry?
- **RQ4.** What are the measurement skills required by the software industry and which of them were acquired in the computer courses?
- **RQ5.** What are the metrics (i.e., metrics for product maintenance, performance, reliability, and other features) covered in the computer science course curriculum?
- **RQ6.** What are the measurment topics (collecting,

 ISBN: 978-1-61208-590-6 

storage, analysis and reporting) covered in the computer science course curriculum?
- **RQ7.** Can the educational game be considered to be appropriate in terms of content relevance, correctness, sufficiency and degree of difficulty, sequence, teaching methodology and duration in the context for which it is intended? Is the game considered to be "engaging"? What are its strengths and weaknesses?
- **RQ8.** How does the effectiveness of learning measurement through gamification compare with that of using traditional learning, in the pre and post questionnaries?

These research questions were defined in an attempt to refute the following null hypothesis:
- **H0.** *There will be no difference in the pre test and post test scores between the two groups (the experimental and control group will have equal skills) when applying the measurement in pratice.*

If the null hypothesis is refuted, we intend to test our alternative hypothesis:
- **H1:** *There will be a difference in pre and post test scores between the two groups (the experimental and control group will not have equal skills) when the measurement strategy is employed in pratice.*

## V. RESEARCH METHOD AND PROGRESS

Since this paper forms a part of a doctoral research, we still do not have sufficient results to effectively answer the research questions; however, all these questions are addressed by discussing the research methods used. The research methods employed to answer the research questions and test the hypothesis will be outlined in this Section.

### A. *Identifying the Diferent Approaches and Benefits of Gamification for the Teaching of Software Measurement*

When answering RQ1 and RQ2, a systematic review of the literature will be carried out to analyze the results, methodologies and tools of the works that are aligned with the subject of gamification and applied to the teaching of software measurement. This systematic review will entail adopting a simplified and adjusted approach from the Kitchenham guidelines [34]. Figure 1 shows the systematic review protocol; the following questions were raised during the the systematic review of the literature (SRLQ) with the aim of finding out about other gamification approaches applied in the area of software measurement. This review will also be used to explore the validation methods applied in other gamification systems and their measurement elements. It should be noted that we are currently working on the systematic review of the literature. The research questions in this review are:
- **SRLQ1.** Based on ISO 15939 where the software measurement features were addressed in the gamification systems?
- **SRLQ2.** In what contexts (i.e., education, work, and other areas) were the measurements for gamification software applied?
- **SRLQ3.** What were the limitations reported in the use of gamification for software measurement education?
- **SRLQ4.** What research methods were used to validate the gamification system?
- **SRLQ5.** What game elements were used in gamification for teaching software measurement?
- **SRLQ6.** What game mechanics were used in gamification for teaching software measurement?
- **SRLQ7.** What game dynamics were used in gamification for teaching software measurement?

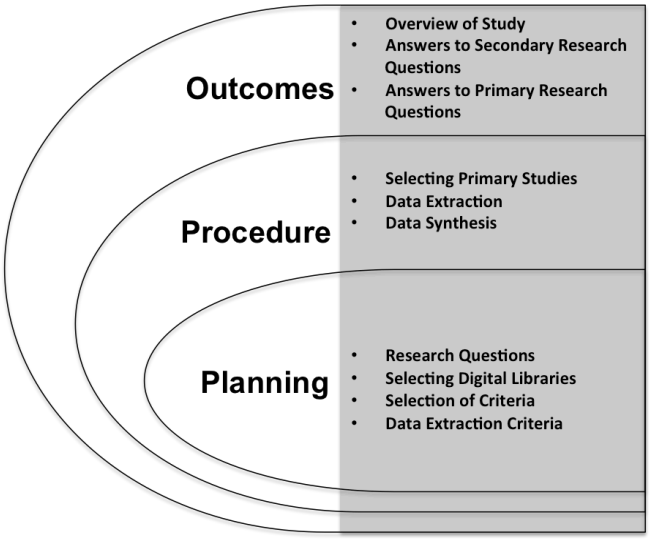


Figure 1. Systematic Review adapted from [34]

### B. *Identifying Measurement Strategies included in Computer Courses and Industry*

With regard to the metrics, indicators and strategies employed in industry, these questions will be investigated in papers that cover this area such as Costa's work [35]. In his work, Costa describes the software measurement process using the Goal–Question–Indicator–Metric (GQIM) methodology [40]. This takes the form of a catalog that lists the following: a) the measurement objectives, b) information needs, indicators and measurements that are most widely used in the context of software process development and c) those identified by a Systematic review of the literature. In addition, a survey will be conducted with applied measurement professionals with a view to answering the RQ3, and thus identifying and giving prominence to the need for Brazilian industry tobe involved in the teaching of software measurement process for undergraduate students.

Moreover, another survey will be carried out to answer RQ4, but this will be conducted with students who have already graduated from the Federal University of Pará and are active in industry. In this way, we will be able to determine which measurement strategies were acquired in an academic environment and which are being used in an

industrial environment. This will also enable us to fill in any gaps with regard to the measurement process in the teaching environment. Both surveys will follow the guidelines recommended by Kitchenham and Pfleeger [36].

Furthermore, when answering RQ5 and RQ6, a literature review will be carried out on the curriculum guidelines of the ACM / IEEE [37] and the SBC [38] to find out which measurement topics and which metrics they cover. This should provide evidence that the measurement activities suggested by the curriculum guidelines meet the requirements of the software industry.

C. *Defining an Approach for Teaching Measurement by Gamification*

After the first six research questions (RQ), have been answered, they will be assessed in terms of the following results:

- The set of measurement practices used in the software industry,
- The recommendations in the curriculum guidelines,
- The current approaches to gamification in teaching software measurement,
- And, in particular, the gap between industry and the academic world with regard to what instrumenst should be used in software measurement.

These results will serve as a framework in the teaching of software measurement by gamification. In addition to the application of these answers, this research will make use of the teaching framework of software measurement found in Villavicencio's work [17], where it introduces gamification concepts into teaching and learning activities. This framework is based on Bloom's taxonomy on levels of learning outcomes [41] and adopts a constructivist approach. The six thinking/learning levels that are defined in Bloom's taxonomy are as follows: recognizing / remembering, understanding, applying knowledge and techniques, analyzing, evaluating, and forming a synthesis. The constructivist approach is based on the assumption that the learners can construct their own knowledge and reach higher levels of learning through an engagement and active participation with it. For this reason, the framework established by Villavicencio is suited to this work since the involvement and commitment of the students is embodied in the learning process. This can be achieved by incorporating the concepts of game elements and mechanics that can be defined in the area of gamification. The Framework can be seen in Figure 2.

D. *Performing Case Studies to Evaluate the Teaching Method*

After the software measurement process by gamification has been implemented, this research will make a comparison of the results obtained from two groups of students as a means of validating and answering RQ7 and RQ8. The two groups will be divided into a control group and an experimental group and comprise Software Quality at the Federal University of Pará, which has approximately 20 to 30 students per class. The control group will not carry out teaching through gamification, in contrast with the experimental group. Thus, the objective results obtained from each class will be taken note of before RQ8 is answered. These results will be analyzed on the basis of the grades achieved at the end of the course, while the subjective results of the classes will be drawn on to answer RQ7. This will be undertaken by setting a post test questionnaire to evaluate the opinion of each student on gamification as a teaching method. These experiments will be conducted over a period of 2 semesters and will follow the guidelines recommended by Wohlin [39]. The hypotheses will be tested with the aid of the data collected in the experimental phase, which include descriptive statistics to analyze the high, low and average grades of both the control group and experimental group. On the basis of this, it will be possible to determine if there is any significant difference between both groups.

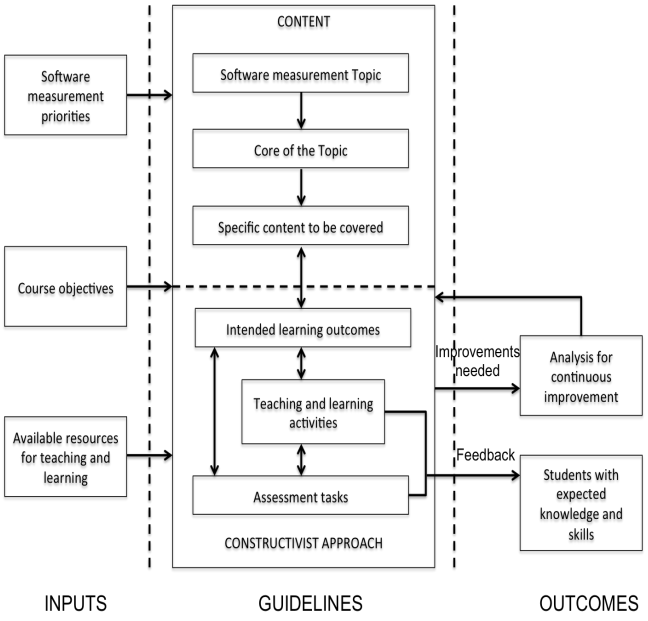


Figure 2. A Framework for software measurement teaching [17]

As the aim of the research is to have an effect on industrial strategies, the survey will have to be conducted in several stages. These will entail the following: a) monitoring the knowledge acquired about the people involved, b) understanding the reality of the students' professional practice, c) undertaking numerous other follow-up research studies, d) taking part in the software development community (SDC) to make use of the improvements in the results and comtribute to them and e) making a practical application of the content in numerous other undergraduate and postgraduate classes.

Finally, it should be emphasized that, to a great extent, this research relies on the quality and quantity of the results obtained from each of the research methods, although the success of each of the methodologies cannot be ensured without first putting them into practice.

 ISBN: 978-1-61208-590-6 

## VI. CONCLUSION AND SUGGESTIONS FOR FUTURE WORK

This paper has underlined the importance of software measurement process, while acknowledging its limitations, especially those aimed at improving teaching practices. In addition, this work forms a part of a doctoral research, which seeks to devise a teaching method of gamification in software measurement and investigate the teaching and learning activities resulting from this.

This involved defining the hypotheses and research questions that will lead to the next stages in this research. In addition, we outlined the research methods that will be used for carrying out this research. These include a systematic review of the literature, surveys, an analysis of curriculum guidelines and the application of a user test conducted at the Federal University of Pará by applying pre and post test questionnaires with students in the subject-area of Software Quality.

In parallel to writing this paper, a systematic review of the literature is being carried out to give an overview of the topic and show the different approaches adopted by the author, while also seeking to answer RQ1 and RQ2. For this reason, the next stage that will be followed will be to conduct a survey to determine the metrics and measurements that are most needed in industry. After this, there will be an analysis of the curriculum guidelines to find out what measurement procedures are needed in the academic world. Thus, as future works, this research will show the state-of-the-art on software measurement education using gamification as an educational tool. It seeks to show the following: i) how educators can benefit from gamification in measurement education and learning, ii) what are the metrics and indicators that are most widely used by the software industry, iii) what are the measurement skills required by the software industry, iv) which of them were acquired in the computer courses, v) what are the metrics (i.e., metrics for product maintenance, performance, reliability, and other factors) covered in the computer science course curriculum, vi) what are the measurement topics (collecting, storage, analysis and reporting) covered in the computer science course curriculum and vii) the most important contribuition - the gamification system to teach software measurement.

This system will be evaluated to determine whether the educational game can be regarded as appropriate in terms of content relevancy, correctness, sufficiency and degree of difficulty, sequence, teaching method and duration, for the purposes to which it is intended. There is also a need to know how the effectiveness of learning measurement with gamification compares with traditional learning methods.

Finally, after all the research questions have been answered, the hypotheses will be tested by the data collected in the experimental phase. This will enable us to determine if there is any significant difference in the learning process of software measurement when the gamification system is applied.

## ACKNOWLEDGMENT

The authors would like to thank the Amazon Foundation for Studies and Research Support (FAPESPA) for awarding a doctoral scholarship to PPGCC/UFPA and the Dean of Research and Postgraduate Studies at the Federal University of Pará (PROPESP/UFPA) by the Qualified Publication Support Program (PAPQ), for the financial support.

## REFERENCES

[1] ISO/IEC, "ISO/IEC 15504-1: Information Technology - Process Assessment - Part 1: Concepts and Vocabulary", Geneve, 2004.

[2] T. Demarco, "Controlling software projects", Yourdon Press Prentice-Hall, 1982.

[3] N. Fenton and S. L. Pfleeger, "Software Metrics. A rigorous and practical approach", PWS Pub, 1997.

[4] M. Kasunic, "The state of software measurement practice: results of 2006 survey", Technical report CMU/SEI-2006-TR-009, Carnegie Mellon University/Software Engineering Institute, Pittsburgh, Pennsylvania, 2006.

[5] C. A. Dekkers and P. A. McQuaid, "The dangers of using software metrics to (Mis)Manage", IEEE IT Professional, IEEE Computer Society, 2002.

[6] C. Jones, "Software Metrics: Good, Bad and Missing", Computer, v.27 n.9, p.98-100, 1994.

[7] E. Herranz, R. Colomo-Palacios, A. de Amescua Seco, and M. Yilmaz, "Gamification as a disruptive factor in software process improvement initiatives", Journal of Universal Computer Science, 20(6), pp. 885–906, 2014.

[8] C. G. Von Wangenheim, C. T. Punter, and A. Anacleto, "Software Measurement for Small and Medium Enterprises - A Brazilian-German view on extending the GQM method", in 7th International conference on Empirical Assessment in Software Engineering (EASE), Keele University, Staffordshire, UK, pp. 1-19, 2003.

[9] O. Pedreira, F. Garcia, N. Brisaboa, and M. Piattini, "Gamification in software engineering, a systematic mapping", Information and Software Technology, 57:157–168, 2015.

[10] J. Iversen and O. Ngwenyama, "Problems in measuring effectiveness in software process improvement: A longitudinal study of organizational change at Danske Data", International Journal of Information Management, vol. 26, pp. 30-43, 2006.

[11] A. Gopal, M. S. Krishnan, T. Mukhopadhyay, and D. R. Goldenson, "Measurement programs in software development: determinants of success", IEEE Transactions on Software Engineering, vol. 28, pp. 863-875, 2002.

[12] M. Díaz-Ley, F. García, and M. Piattini, "Implementing a software measurement program in small and medium enterprises: a suitable framework", Software, IET, vol. 2, pp. 417-436, 2008.

[13] S. Löper and M. Zehle, "Evaluation of software metrics in the design phase and their implication on CASE tools", Master Thesis, Blekinge Institute of Technology, Sweden, 2003.

[14] G. T. Hock and G. L. S. Hui, "A study of the problems and challenges of applying software metrics in software development industry", Proceedings of the M2USIC-MMU International Symposium on Information and Communication Technologies, Putrajaya, Malaysia, pp. 8-11, 2004.

[15] M. Díaz-Ley, F. García, and M. Piattini, "MIS-PyME Software Measurement Maturity Model - Supporting the Definition of Software Measurement Programs and Capabilty Determination", Advances in Software Engineering, vol. 41, pp. 1223-1237, 2010.

[16] J. J. M. Trienekens, R. J. Kusters, M. J. I. M. van Genuchten, and H. Aerts, "Targets, drivers and metrics in software process improvement: results of a survey in a multinational organization", Software Quality Journal, vol. 15, pp. 135-153, 2007.

 ISBN: 978-1-61208-590-6 

[17] M. Villavicencio and A. Abran, “Towards the Development of a Framework for Education in Software Measurement”, Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement, pp. 113-119, 2013.

[18] C. G. Von Wangenheim and M. T. D. Kochanski, “Empirical evaluation of an educational game on software measurement”, Empirical Software Engineering, v.14 n.4, p.418-452, 2009.

[19] R. Solingen and E. Berghout, “The Goal/Question/Metric Method: a practical guide for quality improvement of software development. A Practical Guide for Quality Improvement of Software Development”, New York, McCraw-Hill Publishers, p. 216, 1999.

[20] J. McGarry et al., “Pratical Software Measurement: Objetive Information for Decision Makers”, Addison Wesley, Boston, USA, 2002.

[21] C. Jones, “Software Metrics: Good, Bad and Missing”, Computer, v.27 n.9, p.98-100, 1994.

[22] A. R. Rocha, G. Santos, and M. P. Barcellos, “Software Measurement and Statistical Process Control”, Publication of the Brazilian Ministry of Science, Technology and Innovation, 2012.

[23] M. Villavicencio and A. Abran, “Educational Issues in the Teaching of Software Measurement in Software Engineering Undergraduate Programs”, in Joint Conference of the 21st Int'l Workshop on Software Measurement and the 6th Int'l Conference on Software Process and Product Measurement (IWSM-MENSURA), Nara, Japan, pp. 239-244, 2011.

[24] M. Villavicencio and A. Abran, “Software Measurement in Software Engineering Education: A Comparative Analysis”, in International Conferences on Software Measurement IWSM/MetriKon/Mensura 2010, Stuttgart, Germany, pp. 633-644, 2010.

[25] M. Villavicencio and A. Abran, “Facts and Perceptions Regarding Software Measurement in Education and in Practice: Preliminary Results”, Journal of Software Engineering and Applications, vol. 4, pp. 227-234, 2011.

[26] L. Lindsey and N. Berger, “Experiential approach to instruction”, in Instructional-Design Theories and Models, vol. III, C. Reigeluth and A. Carr-Chellman, Eds., New York: Routledge, Taylor & Francis Group 2009, pp. 117-142. (V1_35), 2009.

[27] AMCHAM, “Only 14% of companies are completely satisfied with the results of IT measurement systems”, 2012, Available: http://www.amcham.com.br, retrieved: March 2017.

[28] C. Jones, “Applied Software Measurement: Global Analysis of Productivity and Quality”, Third ed.: McGraw-Hill Osborne Media, 2008.

[29] A. Abran, “Software metrics and software metrology”, New Jersey: IEEE Computer Society / Wiley Partnership, 2010.

[30] H. Yazbek, “Metrics Support in Industrial CASE Tools”, Software Measurement News: Journal of the Software Metrics Community, 40, 2010.

[31] A. Bartel and G. Hagel, “Gamifying the learning of design patterns in software engineering education”, 2016 IEEE Global Engineering Education Conference (EDUCON), Abu Dhabi, pp. 74-79, 2016.

[32] B. S. Akpolat and W. Slany, “Enhancing software engineering student team engagement in a high-intensity extreme programming course using gamification”, 2014 IEEE 27th Conference on Software Engineering Education and Training (CSEE&T), Klagenfurt, pp. 149-153, 2014.

[33] R. O. Chaves, et al., “Experimental Evaluation of a Serious Game for Teaching Software Process Modeling”, in IEEE Transactions on Education, vol. 58, no. 4, pp. 289-296, 2015.

[34] B. Kitchenham, “Procedures for performing systematic reviews”, Keele, UK, Keele University, v. 33, n. 2004, pp. 1–26, 2004.

[35] T. S. A. Costa, “A Methodological Approach to the Implementation of a Measurement Process based on a Software Tool and a Measurement Tools Catalogue”, Master’s Thesis, Federal University of Pará, pp. 74-127, 2016.

[36] B. Kitchenham and S. Pfleeger, “Personal Opinion Surveys”, in Guide to Advanced Empirical Software Engineering, Springer, 2008.

[37] ACM/IEEE, “Computer science curricula 2013. Curriculum guidelines for undergraduate degree programs in Computer Science”, 2013.

[38] SBC, “Reference curriculum for undergraduate courses in Bachelor in Computer Science and Computer Engineering”, 2005.

[39] C. Wohlin et al., “Experimentation in software engineering: an introduction”, in Kluwer Academic Publishers, Norwell, MA, 2000.

[40] R. E. Park, W. B. Goethert, and W. A. Florac, “Goal-Driven Software Measurement – A Guidebook”, CMU/SEI-96-HB-002 Handbook. Software Engineering Institute, Carnegie Mellon University, Hanscom, MA, pp. 53-59, 1996.

[41] L. Anderson, et al., “A taxonomy for learning, teaching and assessing. A revision of Bloom's taxonomy of Educational Objectives”. Addison Wesley Longman Inc, New York, 2001.

 ISBN: 978-1-61208-590-6 