A tertiary mapping on the use of games for teaching software engineering

Diego Castro, Filipe Arantes, Cláudia Werner
COPPE/Computer Systems Engineering Program
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
{*diegocbcastro, ffernandes, werner*}*@cos.ufrj.br*

***Abstract*—Games are increasingly popular among people. In the academic area, games can be used to give more motivation and engagement to the student, serving as a new teaching method. Software engineering has much content to be taught, due to this, most of the classes in this discipline are theoretical, which entails in a somewhat tiring class. Thus, researchers are increasingly looking for new methods to try to make this teaching more productive and fun. A possible solution to this problem is the use of games as a new method of teaching. Many studies have already produced much material to try to use games as a means of teaching software engineering, demonstrating that games can have a high potential to assist new students in the learning process.**

***Keywords*-Serious Games, Games-Based Learning, Software Engineering, Teaching, Learning, Review, Tertiary mapping**

I. Introduction

There is still a significant difference between Software Engineering (SE) skills taught in a university and skills desired by an organization. The industry is generally unsatisfied with the lack of training of new graduates entering the labor market. This problem seems to derive from how SE is typically introduced to students: general theory is presented in lectures and put into practice in a fictitious project [1].

SE is a discipline that involves interaction, collaboration, and lots of practice. The training of future engineers is something that involves reality and achievement, and students should gain experience, not in isolation, but practicing [2] [3]. The educators should not teach practical applications with just concepts, and it is necessary to have a real application, the student must somehow experience what has been taught in theory so that he/she can learn to use his/her knowledge more productively [2].

Because of this problem of practice, educators are increasingly seeking innovative learning strategies that combine pleasure with education so that problems can be solved by teaching a given subject [3]. One possible solution to make such learning enjoyable is the use of games as a teaching tool. Games are mostly interactive, which is one of the main ways of entertainment and enjoyment, making them an excellent means to make students remember the content in a different and innovative way. Based on this, the use of games in education seeks to make teaching a subject more interactive, more enjoyable, making the game as part of the learning process.

Therefore, to verify the potential of games as a teaching method for SE, a tertiary mapping was performed to identify the secondaries studies that already have been done in the area. The remainder of this paper is presented as follows: Section II describes the research method used in the tertiary mapping, Section III shows the results that were found and Section IV concludes with the final remarks.

II. Research Method

Systematic mapping is a method of secondary study (study that makes aggregation of information based on primary studies related to a specific research question [4]) that systematically (i.e., based on a structured and repeatable process or protocol) explores and categorizes the studies in a given field of research and provides a structure of the types of research reports and results, resulting in an overview on a given subject [4] [5]. The mapping to be presented follows the protocol proposed by Kitchenham [5].

The research process presented in this study covers articles that were published until the end of 2018 and aims to perform a tertiary mapping (mapping of secondary studies to answer a broader research question [4]) to identify the work that has already been performed in the gaming area as a teaching tool for SE. Following will demonstrate the questions research and the implementation procedure used.

Research Questions: **Q1:** What game definition was used? **Q2:** What is the main advantage / motivation of the use of games to teaching software engineering? **Q3:** What is the disadvantage of the use of games to teaching software engineering? **Q4:** What is the main characteristic of the game used? **Q5:** What were the main findings?

Implementation procedure: **(1)** Execute the search string; **(2)** Apply the inclusion criteria based on the title; **(3)**Apply the inclusion criteria based on the abstract; **(4)**Apply the inclusion criteria based on the full text.

Inclusion criteria: **(1)** The article should report a secondary study; **(2)** The article should be in the context of software engineering and games; **(3)** The article must provide data to answer at least one of the research questions; **(4)** The article should be written in English.

Exclusion Criteria: **(1)** Book chapters, conference call; **(2)** Studies that can not be fully accessed; **(3)** Studies that are not in the area of Computer Science or Engineering

A. Search string

The definition of the search keywords, used in the search, was made based on the PICO strategy [6], using three of the four levels. The keywords and structure of PICO can be seen below: **(Population)** Software engineering; **(Intervention)** Tutoring , teach* , instruction , discipline, schooling, educat*, mentoring , course, learn*, train*, syllabus; **(Comparison)** Not applicable; **(Outcome)** Game* , gami*, play*, "serious games", edutainment, "game based learning", simulation.

The search string was defined by grouping keywords of the same domain with the logical operator "OR" and grouping the three domains with the logical operator "AND". In the first instance, the search string formed by the three PICO levels returned a result of 4,428 articles. Because the number of articles returned was relatively large, it was sought to find secondary studies that were ready. To search the secondary studies, a new "AND" was placed at the end of the search string. The search string used can be seen below:

TITLE-ABS-KEY (("software engineering") AND (tutoring OR teach* OR instruction OR discipline OR schooling OR educat* OR mentoring OR course OR learn* OR train* OR syllabus) AND (game* OR gami* OR play* OR "serious games" OR edutainment OR "game based learning" OR simulation) AND ("systematic literature review" OR "systematic review*" OR "mapping study" OR "systematic mapping" OR "structured review" OR "secondary study"))

III. Results

When executing the search string, 36 articles were found, where some criteria of cuts were applied as already explained in the previous section. In the end, 7 documents were left being judged relevant to the work.

Q1: What game definition was used?

In the research that was carried out, it was possible to observe some terms that are directly related to the area of games and teaching, among them: serious games, games based learning, gamification, game development based learning and simulation.

According to Alhammad and Moreno [7], gamification has been considered one of the leading emerging teaching technologies adopted in education. Mauricio et al. [8] define gamification as the use of philosophy, and mechanisms of games in environments other than games to induce specific behavior in people. In other words, everything revolves around a principle, the use of elements of games in other contexts that are not only playful [7].

Although some researchers are confused when using gamification, it is not the same as serious games. Serious games are designed with other purposes, as for example educational, hence the use of the term "serious". They seek some result from the use of games [7], i.e., it can be understood as a game created with another purpose that is not just entertainment. The main difference between gamification and serious games is that games refer to design for non-recreational environments, while gamification refers to employing only the principles and elements of games in a specific process [7].

Finally, there is one more term that is directly connected with the idea of games and teaching; this term is known as simulation. Simulation can be understood as an imitation of the operation of a processor system of the real world over time [9]. According to Caulfield et al. [10], a game is different from a model or simulation. A model is a representation of a complex reality that reflects specific selected characteristics of the system it represents. It is useful in that it accurately portrays those features that are interesting at the moment. Meanwhile, a simulation is a type of model that displays processes in some way and shows how these processes change from state A to state B. A game is a simulation that is purposely executed, totally or partially determined by the decisions of the players, within certain predetermined circumstances.

Q2: What is the main advantage/motivation of the use of games to teaching software engineering?

Games have some features that can be used in teaching to make it more efficient, such as the fact that the game captures the user's attention, ability to practice, and fun. These characteristics, when used correctly, can bring some advantages to teaching.

Alhammad and Moreno [7] reported that most of the studies they found affirmed that gamification could have a positive impact on increasing user motivation and engagement about a particular behavior. However, the usefulness of gamification depends on the context in which it was applied.

Game-related educational methods have been used primarily to minimize the gap between theory and practice [11]. Serious games have been recurrent tools to improve the learning process and engagement of SE students [11], the primary purpose of the lesson is to learn while the students are playing, thus making fun learning.

From the study conducted by da Silva et al. [12], it was possible to observe that 97% of the works analyzed by them had a great result in the introduction of games as a teaching method, among the main benefits cited by the authors are: gain of motivation, increase of concentration, decrease in the number of disengagement of the course, and increase of knowledge in the matter. According to Kosa et al. [13], game motivational aspects gain importance in Software Engineering Education (SEE) because software engineers need to work in a highly social and collaborative environment and games are good at providing the necessary collaboration while being played and designed, thus bringing

benefits both to GBL and GDBL.

Simulation increases the opportunity to learn from failure without loss in reality, but in a very timely manner and, furthermore, still manages to bring motivation and engagement on the part of students. According to them, one of the main advantages that simulation can bring is feedback, that in this way the student knows if a particular task is being done correctly or not, seriously influencing the effects of learning [9].

Finally, Caulfield et al. [10] give importance to other advantages offered by games: the ability of repeatability; a game is a visible and physical representation of a problematic space; a captured mental model; there are places to test new ideas and experiment with established theories; areas where time and space can be contracted or expanded; places where it is acceptable to try different things and learn through failures [10].

Q3: What is the disadvantage of the use of games to teaching software engineering?

Even though there are not many problems, the use of games in education may also present some drawbacks when used, based on this study, it was possible to identify some of them.

Alhammad and Moreno [7] argued that gamification is still new and immature, and so one should be very careful when trying to gamify something. Many applications do not achieve their business objectives, due to the introduction of many elements at the same time, according to the authors there are cases where gamification has not been gradually incorporated, and negative results were obtained. The rapid insertion may have caused a feeling of confusion or frustration that prevented students from understanding and enjoying the "game". A gamification approach should be introduced gradually, starting with no new activity and only integrating gamification elements into existing processes [7].

Mauricio et al. [8], Souza et al. [11] and Caulfield et al. [10] suggest that while games are pedagogical tools and are well received by players, they are not sufficient by themselves and should be complemented by other learning devices. That is, the serious game can not alone teach something; it must be supported by theory; it works more as a strengthening of knowledge.

Finally, Caulfield et al. [10] add that there are some dangers in using games as a teaching method. Games are just a representation of how the world works, so it is potentially dangerous to have players leaving the game environment with the belief that the strategies that were employed in the game are directly transferable to the real world.

Q4: What is the main characteristic of the game used?

SE has difficulties in demonstrating some concepts through lectures and traditional classes, given the limitations of these formats, various concepts presented in this discipline need a practical application. Based on this, many scholars are already using games to be used in their courses to assist in the current teaching method [8].

Several games to teach software engineering were found, following will show the main games that were found grouped by educational area: **Software Process Management:** ARMI, eRisk, SimSE, Problems and Programmers, SimSoft, Software Hut, PlayScrum, SimVBSE, SimjavaSP, MO-SEProcess, X-MED; **Specification:** Groupthink; **Software Test:** Pex4Fun, CodeHunt; **Software Design:** ViRPlay3D; **Modeling:** SESAM; **Software Requirements:** Software Quantum Game [8].

Alhammad and Moreno [7] identified four characteristics that games present, the first being the increase of student involvement. Then it was presented that they improved knowledge or performance. The third characteristic was that they encourage the use of the best practices of SE and, lastly, they developed the teamwork and social skills of the students. Da Silva et al. [12] reported some competences that can be explored through the use of games depending on their main characteristics. The main competencies found were: logical conclusions, finding errors through problems to be solved, development, responsibility, curiosity, confidence, teamwork, work with rules, creativity and work in a structured way.

Mauricio et al. [8] have been able to identify some characteristics and objects that games can have, among them: classification tables, points, levels, progress, competition, collaboration, teams, rewards, challenges, stories, achievements, time pressure, missions, badges, feedback and market. Other features that may exist are: multiple players, different roles in the game, different experiences, different purposes for fun, different levels, simulation environment, collaboration, competition, different interaction environments, scores, end result, practical failure analysis, particular teaching purpose, make decisions, realistic scenario and time [9] [7].

Q5: What was the main finding (s)?

Alhammad and Moreno [7] sought to find out which were the major areas of SE that were being addressed using games as teaching method. From this study, twenty one papers were found. Of this total, seven articles sought to teach software process, four tried to teach approaches to software building, three talked about process improvements, and two dealt with software testing problems. There were also papers describing others subjects, such as: software configuration, requirements, design, risk, and maintenance, all of these with only one paper.

In the study, some approaches to game-based teaching were found, according to the survey by Mauricio et al. [45], in a total of 159 reviews, 88 presented the use of GBL method, 60 showed the use of GDBL method, and 11 provided Gamification approaches. Three studies showed hybrid approaches, that is, the combination of two distinct ways: one used Gamification and GDBL, and two used GDBL and GBL, so they were accounted for in the two approaches that were used. Therefore, GBL and GDBL

methods are the most used currently to aid teaching.

According to Souza et al. [11], in a total of 106 articles, 57 studies presented GBL approaches, 38 presented GDBL approaches, 8 presented Gamification approaches, and 3 presented hybrid approaches, strengthening once again that GBL and GDBL are the most used approaches.

GDBL and GBL have characteristics that make these approaches the most used and still have distinct characteristics that make each one more appropriate to a specific context or objective. The most significant difference identified is that GBL is generally restricted in a shorter duration and provides simplifications that encourage the perception of specific skills or details on a given topic [11]. This feature makes GBL more suitable for specific learning objectives and to give a deeper understanding of particular topics. On the other hand, GDBL leverages problem-based learning, presenting students with challenges in which they often need to apply SE techniques. By applying SE best practices, students discover the inherent problems and challenges of the discipline themselves [11]. Therefore, GDBL is best suited for extensive learning goals and for the development of skills through practice.

IV. Final Remarks

SE is a discipline that addresses valuable solutions to complex problems, which requires the right combination of technical and practical information. However, typical education in SE with theoretical background could not highlight the possible practical issues with which a software engineer should deal directly. SE educators are continually looking for effective ways to address this challenge. For this, games can be considered as a tool or a solution to train novice software professionals and can still solve problems that cause a lack of practical information.

SE cannot be taught exclusively in the classroom, as it is a competence, not just a body of knowledge, and presentation of principles and experiences, which is not supported by the active and regular participation of students in projects, students will undoubtedly fail to realize the essence of what the learner needs to learn.

Taking into account that games can help in learning due to its main characteristics such as motivation, a gain of attention, fun, and practice, Caulfield et al. [10] state that games are already being used in SEE, and students generally like to feel some value from experience. However, few games have been developed beyond their initial implementations, suggesting that their pedagogical value has not been sufficiently demonstrated and thus more is expected.

References

[1] E. O. Navarro, A. Baker, and A. Van Der Hoek, "Teaching software engineering using simulation games," in *Proceedings of the International Western Simulation Multiconference*, 2004.

[2] S. Hadjerrouit, "Learner-centered web-based instruction in software engineering," *IEEE Transactions on Education*, vol. 48, no. 1, pp. 99–104, 2005.

[3] U. Ritterfeld, M. Cody, and P. Vorderer, *Serious games: Mechanisms and effects*. Routledge, 2009.

[4] S. Keele *et al.*, "Guidelines for performing systematic literature reviews in software engineering," Technical report, Ver. 2.3 EBSE Technical Report. EBSE, Tech. Rep., 2007.

[5] B. Kitchenham, "Procedures for performing systematic reviews," *Keele, UK, Keele University*, vol. 33, no. 2004, pp. 1–26, 2004.

[6] M. Petticrew and H. Roberts, *Systematic reviews in the social sciences: A practical guide*. John Wiley & Sons, 2008.

[7] M. M. Alhammad and A. M. Moreno, "Gamification in software engineering education: A systematic mapping," *Journal of Systems and Software*, vol. 141, pp. 131–150, 2018.

[8] R. d. A. Mauricio, L. Veado, R. T. Moreira, E. Figueiredo, and H. Costa, "A systematic mapping study on game-related methods for software engineering education," *Information and software technology*, vol. 95, pp. 201–218, 2018.

[9] S. Jiang, H. Zhang, C. Gao, D. Shao, and G. Rong, "Process simulation for software engineering education," in *Proceedings of the 2015 International Conference on Software and System Process*. ACM, 2015, pp. 147–156.

[10] C. Caulfield, J. C. Xia, D. Veal, and S. Maj, "A systematic survey of games used for software engineering education," *Modern Applied Science*, vol. 5, no. 6, pp. 28–43, 2011.

[11] M. R. Souza, L. Veado, R. T. Moreira, E. Figueiredo, and H. Costa, "Games for learning: bridging game-related education methods to software enginering knowledge areas," in *Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track*. IEEE Press, 2017, pp. 170–179.

[12] T. R. da Silva, T. J. Medeiros, and E. H. da Silva Aranha, "The use of games on the teaching of programming: A systematic review," in *Proceedings of the Workshop on Experimental Software Engineering (ESELAW'15)*, 2015, pp. 474–487.

[13] M. Kosa, M. Yilmaz, R. O'Connor, and P. Clarke, "Software engineering education and games: a systematic literature review," *Journal of Universal Computer Science*, vol. 22, no. 12, pp. 1558–1574, 2016.