Games and Gamification in Software Engineering Education: A Survey with Educators

Pedro Rodrigues
Computer Science Department
Federal University of Minas Gerais
Belo Horizonte, Brazil
pcampolina@gmail.com

Mauricio Souza
Computer Science Department
Federal University of Minas Gerais
Belo Horizonte, Brazil
mrasouza@dcc.ufmg.br

Eduardo Figueiredo
Computer Science Department
Federal University of Minas Gerais
Belo Horizonte, Brazil
figueiredo@dcc.ufmg.br

***Abstract*—The use of games and game elements in software engineering education is not new. In fact, their use in Software Engineering education is found in research papers since 1974, with a notorious increase after 2000. However, there is little information about the actual adoption of these approaches in software engineering education. Therefore, the goal of this paper is to investigate the use of games and game elements in software engineering education, in the perspective of educators. To achieve this goal, this study proposes and analyzes the results of a survey answered by 88 software engineering professors. We sample the participants by inviting 285 educators mined from one hundred well-stablished universities and educational institutions of different regions of Brazil. The goal of the survey is (i) to collect information about the use of games and gamification in classrooms and (ii) to understand the relation of ACM/IEEE knowledge areas and the used game-related methods. The results show that most of the professors are aware of these educational approaches, the games were adopted by only 21 participants and game elements were only adopted by 19 participants. Games are most used to cover "Software Process" and "Project Management". The most used game elements are Points, Quizzes, and Challenges. The results also show that the main reasons for not adopting the resources are the lack of knowledge, lack of information about relevant games for teaching software engineering, and the lack of time to plan and include these approaches in the classroom. Finally, results show a positive tendency towards the future adoption of these game-related approaches by the software engineering professors.**

Keywords— Serious games, game elements, gamification, software engineering education, survey.

I. INTRODUCTION

Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software [1] [10]. The challenges of educating new software engineers is more than just programming, they include attention to details, such as quality, schedule, and economic goals [1] [19]. For instance, an important challenge in Software Engineering education arises from the dual nature of the Software Engineering discipline: it has roots in computer science and has emerged as an engineering discipline, and it affects both theory and practice [1]. This characteristic has a direct impact on the amount of material instructors must cover in Software Engineering classrooms. In addition, software professionals are required not only to understand technical challenges but also to be up-to-date with nontechnical issues, including management, communication, and teamwork.

As a consequence, it is necessary to move beyond the expository classroom format, since it does not favor effective student learning [7] [9] [14]. One approach that has been used to overcome these challenges is the adoption of game-related approaches for software engineering [2] [19] [21].

Souza et al. (2018) identifies three categories of game-related approaches used in software engineering education: Game-Based Learning (GBL), which deals with the use of serious games to support learning; Game Development Based Learning (GDBL), which is the use of game development projects for hands-on experience in the learning process; and Gamification, which is the use of game elements and game-design techniques in non-gaming contexts [7] (i.e., in education, in this case).

However, the selection of educational methods and approaches in learning process is influenced by several criteria, such as the flexibility and ease of using the approaches, their suitability for being used by most instructors, and the effort, restrictions and skills involved in the use of these approaches [13].

Therefore, the goal of this study is to understand the level of adoption of Game-based Learning and Gamification by software engineering professors. We want to understand the reasons and motivations for using or not using these tools in classroom. To achieve this goal, we analyze the results of a questionnaire sent to 285 software engineering professors from 100 renowned superior education institutions from Brazil. As a result, 88 professors participated in the study.

The data obtained show that both approaches are known by most of the participants, however only a fraction of them have already applied these approaches. Serious games have been most used to cover "Software Process", "Project Management" and "Software Requirements" knowledge areas. In the case of Gamification, the most used game elements are Points, Quizzes, and Challenges. The main reasons for not using these approaches are related to lack of knowledge, lack of information about relevant games for teaching software engineering, and the lack of time to plan and include these approaches in the classroom. Finally, there is a positive tendency toward the future adoption of these game-related approaches by the participants.

The remainder of this paper is organized as follows. Section II provides the necessary theoretical foundation for the study.

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

Section III describes the study design, research questions and methods. Section IV presents the results of the study. Section V discusses the main findings and implications of the study. Section VI discusses the possible bias that threatens the validity of this study. Section VII presents the related work. Finally, Section VII presents our conclusions and future works.

II. BACKGROUND

This section provides the theoretical foundation for the topics addressed in this paper. Section II-A discusses the use of Serious Games in software engineering education. Section II-B discusses the use of Gamification in software engineering education.

A. Serious Games for Software Engineering Education

Serious games are game applications that have defined learning outcomes [18] [24]. Games are "any contest (play) among adversaries (players) operating under constraints (rules) for an objective (winning, victory, or pay-off)". Therefore, GBL approaches apply games with the purpose of learning specific skills and concepts, usually named as "serious games" (games with purposes), edutainment or educational games. In this study, we do not limit this concept to digital games, or to the use of serious games: commercial games (games with the pure intent of entertainment) used in the context of learning software engineering are also considered.

The use of games as learning tools is not new and is a common approach to provide variety in teaching and learning approaches [14] [18] [18] [21] [23]. Serious games are specially used as complementary tools for addressing specific learning outcomes, rather than as a substitute for traditional classes.

Examples of Serious Games for learning software engineering include (i) Problems and Programmers [5] and (ii) SimSE [14]. The first (i) is a non-digital card game that simulates software development process from conception to completion. The players compete to finish their projects while avoiding the potential pitfalls of software engineering. The second (ii) is a computer simulation-based game whose goal is to allow students to practice a "virtual" software engineering process (or sub-process) in a fully graphical, interactive, and fun setting. It provides direct, graphical feedback that enables students to learn the complex cause and effect relationships underlying the processes of software engineering.

B. Gamification in Software Engineering Education

Gamification is a recent term in literature and defines the use of game elements and game-design techniques in non-gaming contexts [2] [7]. It was rapidly incorporated in software engineering education as a strategy to motivate students in performing desired behaviors, such as the more frequent use of specific tools, acquiring the habit of applying specific techniques, or being more participative in the classroom [21]. Rather than being a proper learning tool or directly supporting the development of new skill, Gamification is used as a strategy to induce learners to use specific software engineering abilities or practices, by promoting competition or systematically rewarding learners as they perform expected actions or show expected behaviors [21] [25].

For instance, Akpolat and Slany [3] introduced Gamification in an educational software engineering project, where there are weekly challenges to motivate students on applying eXtreme Programming practices to their project and the students compete for a "challenge cup" award. Uskov and Sekar [22] proposes the incorporation of over 20 Gamification elements in modern software engineering courses. The authors organize Gamification elements in three categories: i) Progression Gamification Techniques; ii) Feedback Gamification Techniques; and iii) Behavior Gamification Techniques.

III. STUDY SETTINGS

This section explains how we planned and executed this study. Section III-A presents the study goal and research question. Section III-B discusses the research method we adopted. Section III-C describes the questionnaire and the process used to create it. Section III-D describes our target population and the strategy used to invite participants for the study.

A. Study Goal and Research Questions

The goal of this study is to investigate the adoption of Serious Games and Gamification in the context of software engineering education from the perspectives of professors of top higher education institutions in Brazil. To achieve this goal, we formulated three Research Questions (RQ) presented below:

RQ1. Do software engineering professors use Game-based Learning to support software engineering education?

RQ2. Do software engineering professors use Gamification to support software engineering Education?

RQ3. What are the reasons for not adopting Serious Games and Gamification in software engineering education?

Therefore, for RQ1 and RQ2 our interest is (i) to collect information about the use of games and gamification in classrooms by Software Engineering professors and (ii) to understand the relation of ACM/IEEE knowledge areas and the game-related methods. For RQ3 our goal is to understand what reasons avert educators from using these approaches.

B. Study Design and Research Methods

To answer the research questions, we performed an Opinion Survey study. According to Easterbrook et al. [8], Survey studies are used to identify characteristics of a wide population and are usually associated to the application of questionnaires. Surveys are meant to collect data to describe, compare or explain knowledges, attitudes and behaviors [16].

Therefore, the following steps were defined for the execution of this study: (i) Definition of study goals and research questions (Section III-A); (ii) Creation of the questionnaire (Section III-C); (iii) Execution of a pilot study to validate the questionnaire (Section III-C); (iv) Invitation of the target population to participate in the study (Section III-D); (v) Data normalization, analysis and report of results (Section IV).

C. Questionnaire

To answer the research questions, we created an electronic survey using Google Forms. The survey had five sections: (1) general information about the participants, (2) questions about software engineering education, (3) the utilization of game-based approach in software engineering education, (4) the utilization of Gamification in software engineering education and (5) the reasons of professors for not using the games and gamification in software engineering education. We used multiple-choice and open questions in the questionnaire.

We performed a pilot study with 10 software engineering professors. From the feedbacks of the pilot study, only minor corrections were made. Tables I, II, III, IV and V present the questions in the final version of the questionnaire. Table I presents the questions about the participants background.

TABLE I. BACKGROUND QUESTIONS.

ID	Questions
Q1	What is the actual or recent university or institution in which you have taught Software Engineering? (Open answer)
Q2	Age range (multiple choice) ( ) Up until 30 years old ( ) 31 - 50 years old ( ) More than 50 years old
Q3	Gender (multiple choice) ( ) Male ( ) Female

TABLE II. EXPERIENCE IN SOFTWARE ENGINEERING EDUCATION

ID	Questions
Q4	What disciplines do you teach or have taught in the last years? (check boxes) [ ] Software Engineering 1 or equivalent [ ] Software Engineering 2 or equivalent [ ] Specific discipline in Software Engineering (ex: Software quality or Software reutilization)
Q5	If you teach specific disciplines in Software Engineering, list which ones. (Open answer)
Q6	How long have you been teaching Software Engineering disciplines? (multiple choice) ( ) 0 - 5 years ( ) 5 - 15 years ( ) More than 15 years
Q7	To what intensity do you cover these software engineering areas in your classrooms? Software analysis: ( ) 1 ( ) 2 ( ) 3 Software modeling: ( ) 1 ( ) 2 ( ) 3 Software requirements: ( ) 1 ( ) 2 ( ) 3 Software architecture: ( ) 1 ( ) 2 ( ) 3 Software verification and validation: ( ) 1 ( ) 2 ( ) 3 Software process: ( ) 1 ( ) 2 ( ) 3 Software management: ( ) 1 ( ) 2 ( ) 3 Software quality: ( ) 1 ( ) 2 ( ) 3 Professional practices: ( ) 1 ( ) 2 ( ) 3 Software development: ( ) 1 ( ) 2 ( ) 3 I3 Where: 1 represents “I don't cover this area”; 2, “I cover superficially this area”; and 3, “I extensively cover this area”.

Table II presents the questions about the Software Engineering teaching. In Q4, by defining the options “Software Engineering 1” (SE 1) and “Software Engineering 2” (SE 2) we attempted to condensate the many variable names of national Software Engineering disciplines and separate them in the basic and advanced options. The pilot study and the final results show that these options were understood by the participants. In Q7, the areas shown for the participants are an adaptation of the areas defined by ACM and IEEE curriculum guidelines [1].

Table III presents the questions related to the utilization of game-based approach in Software Engineering teaching. The questions Q9, Q10, Q11 and Q12 were only reached by the participants that answered “Yes, and I have already used in my classrooms” in Q8.

TABLE III. USE OF GAMES IN SOFTWARE ENGINEERING EDUCATION.

ID	Questions
Q8	Do you know the game-based learning approach? (multiple choice) ( ) Yes, and I have already used in my classrooms ( ) Yes, but never used ( ) I do not know
Q9	What games have you already used in your classrooms? (Open answer)
Q10	What areas of Software Engineering teaching you pretended to cover by using the game-based approach? (checkboxes) [ ] Software analysis [ ] Software modeling [ ] Software requirements [ ] Software architecture [ ] Software verification and validation [ ] Software process [ ] Software management [ ] Software quality [ ] Professional practices [ ] Software development
Q11	How do you grade (from 0 to 5) the students acceptance of the games used? In the below scale, 0 represents “Really bad acceptance, many problems” and 5 represents “Great acceptance, no problems” (0 to 5 scale)
Q12	What were the benefits and the difficulties observed about the game-based techniques used in classrooms? Do you identified any performance improvement by the students? (Open answer)

Table IV presents the questions related to the utilization of Gamification approach in Software Engineering teaching. The questions Q14, Q15 and Q16 were only reached by the participants that answered “Yes, and I have already used in my classrooms” in Q13. Finally, Table V presents questions on the reasons why professors do not use games and Gamification in software engineering education.

D. Population and Sampling

The target population of this study are software engineering professors from higher education institutions in Brazil. To select an appropriate sample of this population, we used a ranking of the top one hundred well-established universities and educational institutions of different regions of our country, which is updated yearly by a relevant magazine (http://ruf.folha.uol.com.br/). We searched the websites of each institution for the professors associated to departments or undergraduate programs related to Software Engineering, Computer Science and alike. Finally, we searched the curriculum of each professor in the Lattes Platform [11], which is the official curriculum platform for higher education

professors and researchers in our country, and then we filtered the professionals relevant to our sampling. This step was very important to guarantee that all invitation e-mails were sent to active Software Engineering professor.

For each contact gathered, we sent a personalized invitation e-mail to participate in our study. The idea of the personalized email was for the recipient not confusing the invitation with spam and marketing e-mails, and for the participant to notice he was objectively picked because of his relevance in our sample. This particular strategy worked very well: for the 285 invitations sent, we had 88 answers in our survey (30.9%). This participation rate exceeded our expectations, considering that the target population was very specific. Therefore, we believe our sample is significant for our target population.

TABLE IV. USE OF GAMIFICATION IN SOFTWARE ENGINEERING EDUCATION

ID	Questions
Q13	Do you know the game-based learning approach? (multiple choice) ( ) Yes, and already used in my classrooms ( ) Yes, but never used ( ) I do not know
Q14	What games elements have you already used? (checkboxes) [ ] Points [ ] Badges [ ] Leaderboards [ ] Quizzes [ ] Levels - different difficulties [ ] Challenges [ ] Milestones [ ] Other (open field)
Q15	How do you grade (from 0 to 5) the students acceptance of the games used? In the below scale, 0 represents "Really bad acceptance, many problems" and 5 represents "Great acceptance, no problems" (0 to 5 scale)
Q16	What were the benefits and the difficulties observed about the game-based techniques used in classrooms? Do you identified any performance improvement by the students? (Open answer)

TABLE V. QUESTIONS ABOUT THE REASONS FOR NOT USING THE APPROACHES AND THE PERSPECTIVE OF FUTURE USE.

ID	Questions
Q17	What are the reasons for you never had used games or games elements in Software engineering teaching? (Open question)
Q18	Do you pretend to use either games or games elements in Software Engineering teaching in the future? (Open question)

IV. RESULTS

From 285 invitations sent, we collected 88 responses. All of the 88 responses were validated and entered in the final results analysis. The sample was composed of professors from 45 different institutions. There are 60 male participants (68.2%) and 28 female participants (31.8%). Regarding the age of the participants, 7 (8%) are aged up to 30 years old, 64 (72.7%) between 31 and 50, and 17 (19.3%) are more than 50 years old. Finally, regarding their experience in teaching software engineering, 37 participants (42%) teach software engineering from 5 to 15 years, 28 (31.8%) for more than 15 years, and 23 (26.1%) for less than five years.

Figure 1 shows the distribution of the answers regarding the disciplines taught by the participants (Q4). Considering that Software Engineering is a very diverse and dense area in Computer Science, we tried to resume Software Engineering in two basic categories, one for basic and other for advanced subtopics: SE 1 (introduction) and SE 2 (advanced topics). In addition, we included the option "Specific disciplines", that aimed subtopics of Software Engineering explored to a level where a full discipline was created and used to teach this content. Seventy-one participants (80%) declared that they teach "Specific disciplines", 55 participants (62.5%) declared that teach SE 2, and 34 participants (38.6%) declared that teach SE1.

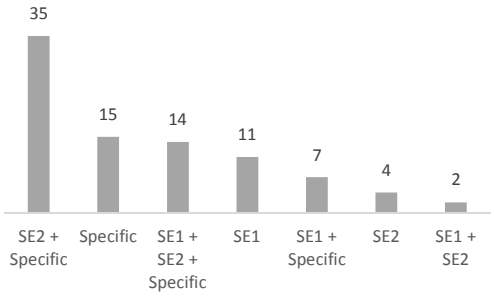

Fig. 1. Disciplines taught by the participants

Table VI shows the most recurrent specific disciplines cited in the answers of Q5. It was an open field, so, in many cases, multiple disciplines were listed. Other 35 unique specific disciplines were cited, what shows the diversity and density of Software Engineering.

TABLE VI. SPECIFIC DISCIPLINE MENTIONED IN Q5.

Specific Discipline	# occurrences	%
Software Project Management	29	23.2
Software Testing	11	8.8
Software Requirements	8	6.4
Software Quality	8	6.4
Software Architecture	8	6.4
Object Oriented Programming	6	4.8
Software Verification and Validation	4	3.2
Software Development	4	3.2
Experimental Software Engineering	4	3.2

Figure 2 shows the coverage of the software engineering knowledge areas (Q7), where values 1, 2 and 3 represent, respectively, "I don't cover this area", "I cover superficially this area" and "I extensively cover this area". By far, "Professional practices" is the least covered area withe "Software process", "Software modeling" and "Software requirements" are the top-3 most covered areas.

A. *The use of Game-based Learning in software engineering education (RQ1).*

Regarding the participants knowledge and use of Game-based learning (Q8), 54 participants (61.4%) know the approach but never used it, 21 participants (23.9%) have already used this approach, and only 13 participants (14.8%) have no knowledge on the subject.

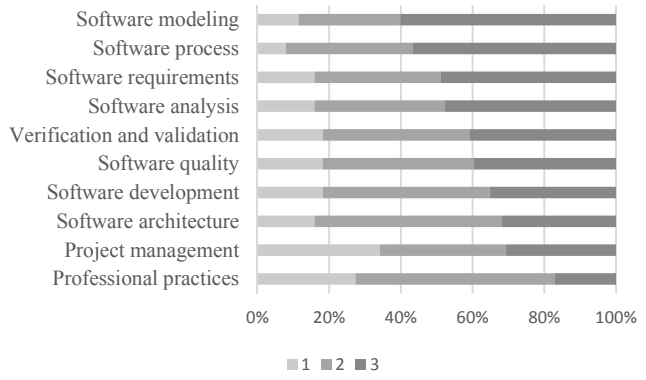

Fig. 2. Software engineering knowledge areas covered by the participants.

The participants that claimed to have used Game-based Learning mentioned 24 games as responses for Q9. Only SimSE [14] and UbiRE [12] had more than one occurrence. Other examples of games include SimSE SPIAL, Code Defenders, Dojo, Sesam, u-Test, JoVeTest, InspSoft, iTestLearning, ScrumGame, XPGame, Requirement Island, Airplane Factory, and other games with generic names such as "Elicitation Theater" and "Puzzle". Figure 3 shows the software engineering knowledge areas the participants used games to teach (Q10). Project Management, Software Process and Software Requirements are the top 3 knowledge areas covered with games, while Software Architecture, Software Maintenance and Software Modelling are the least covered topics.

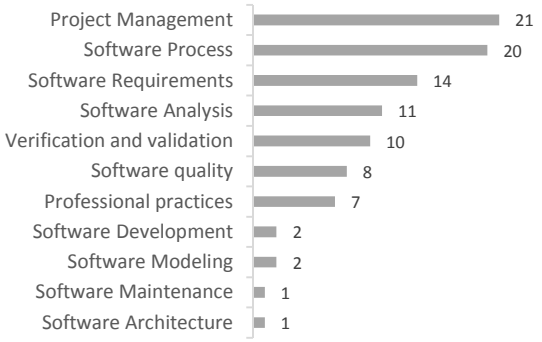

Fig. 3. Software engineering knowledge areas covered by the games

The participants were asked to rate the acceptance of the approach by students and the overall success of its use in their classroom (Q11). This impression was asked in a grade form, where 0 meant "Really bad acceptance, many problems" and 5 meant "Great acceptance, no problems". Figure 4 shows there was an overall positive response for this question, with only grades higher or equal to 3.

Finally, Tables VII and VIII list the main benefits and drawbacks observed in the use of game-based learning (Q12). The benefits were related to keeping students interested, engaged and motivated, improvement in students understanding and assimilation of contents, increased student participation, increased grades and the novelty of the approach. There was no consensus on the difficulties of using game-based learning, and 12 different issues were pointed by the participants.

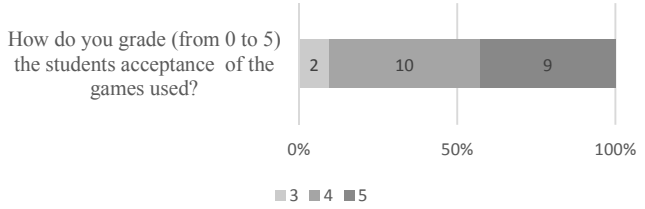

Fig. 4. Reception of Game-based Learning by the students in the perspective of the participants

TABLE VII. BENEFITS OBSERVED FROM THE USE OF GAME-BASED LEARNING.

Benefits	# Occurrences
Higher students interest	7
Higher students engagement	4
Higher students motivation	2
Higher students content assimilation	2
Possibility of teamwork	1
Higher understanding of concepts	1
Higher classroom participation	1
New knowledge approach	1
Higher students grades	1

TABLE VIII. DRAWBACKS OBSERVED FROM THE USE OF GAME-BASED LEARNING

Drawbacks	# Occurrences
Games need a specific learn rhythm	1
Students resistance about the benefits of using games	1
Extra workload for professor	1
Students interest decreases with time	1
Language	1
Game comprehension	1
Applying the technique in a short period of time	1
Competitiveness in excess	1
Higher discipline complexity	1
Games are old and outdated	1
Implementation in classroom	1
Good student immersion is required	1

B. *The use of Gamification in software engineering education (RQ2).*

Regarding the participants knowledge and use of Gamification in software engineering education (Q13), similar to the responses of Q8 (Section IV-A), the majority of the participants (56 participants – 63.6%) know the approach but never used it, 19 participants (21.6%) have already used this approach, and only 13 participants (14.8%) have no knowledge on the subject.

Table IX presents the elements of games used by professors that claimed to have used game elements or Gamification (Q14). Points, Levels and Quizzes were the most mentioned game elements. However, Contests, Badges, Levels and Leaderboards are also mentioned significantly. The literature on Gamification in software engineering suggest that Points, Levels and Milestones are the most common elements used [21].

TABLE IX. GAME ELEMENTS MENTIONED IN THE ANSWERS OF Q14.

Game elements	# Occurrences
Points	15
Challenges	13
Quizzes	11
Badges	7
Milestones	7
Levels	7
Contests	7
Leaderboards	1
Compensation	1
Balanced challenge	1
OC2-RD2 Scripting	1
Logical games	1
Lateral thinking	1
Detailed feedback	1
Sorted questions	1

Figure 5 shows the distribution of responses for Q15. The impressions of the 19 participants who have used this approach regarding the students' acceptance of gamification are positive. However, two participants answered the values "0" and "1" indicating the success of this approach is not a consensus.

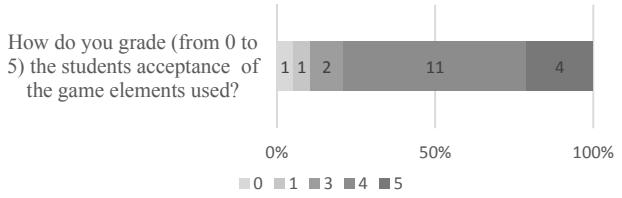

Fig. 5. Acceptance of gamification by the students

Finally, Tables X and XI presents the benefits and drawbacks of using gamification in software engineering education (Q16). Improving students' interest in learning is the dominant benefit described by the participants. For the difficulties, the main issue is objectively measuring any improvement in performance from the students. Four issues are related to the difficulties in executing the approach (higher effort for the professor, lack of tool support, difficulty to adapt the approach to the learning process, and the difficulty in providing instant feedback). The other two problems were related to the students' reception.

TABLE X. BENEFITS OBSERVED FROM THE USE OF GAMIFICATION

Benefits	# Occurrences
Higher students interest	8
Higher students participation	1
Higher students comprehension of content	1
Easier to identify students that don't work in team projects	1
Higher students performance	1

TABLE XI. DRAWBACKS OBSERVED FROM THE USE OF GAMIFICATION

Difficulties	# Occurrences
Difficulty in measuring performance improvements	3
Higher workload to the professor	1
No digital platform to apply the technique	1
Difficulty in reaching all students	1
Students have difficulty in understanding the approach	1
Some students didn't like it	1
Introducing game elements in teaching	1
Difficulty in providing instant feedback	1

C. *Reasons for not using Game-based Learning and Gamification in software engineering education.*

The participants who responded that had never used the approaches (Q8 and Q13) were asked the reasons for not using these approaches (Q17). Table XII list the results for this question. The lack of knowledge about the mentioned approaches is the most recurring reason pointed. The answers related to "Lack of knowledge of appropriate games", "Lack of materials" and "Lack of resources" correspond to 20 out of 63 (31.7%) of the reasons, and these answers may be related to the difficulty in finding relevant guidelines or centralized resources to support the use of these approaches.

TABLE XII. REASONS FOR NOT USING GAME-BASED LEARNING AND GAMIFICATION IN SOFTWARE ENGINEERING EDUCATION.

Reasons	# Occurrences
Lack of knowledge about the approaches	19
Lack of knowledge of appropriate games	14
Lack of time	12
Don't believe in the approaches	6
Lack of interest	5
Lack of material	4
Lack of resources	2
Difficulty in ensuring correct use by students	1

Finally, the participants were asked if they would consider using these approaches in the future (Q18). In our sample, only 12 participants stated that had already used both approaches. Removing these participants, we have a total of 76 responses. In this group, 35 participants (46.1%) were positive about using these approaches in future, 30 participants (39.5%) were unsure, and 7 participants (9.2%) stated that would not use. Four participants (5.3%) did not answer the question. The 7 participants that stated they would not use these approaches in future have never used neither games nor game elements. Therefore, there is a positive tendency towards the adoption of games and game elements in software engineering education.

V. DISCUSSION

In this section we discuss some relevant findings and issues observed in the results of Section IV, and we revisit the Research Questions defined in Section III.

A. *Do professors use game-based learning in education?*

Our study shows that a significant amount of software engineering professors uses games to support software engineering education. Seventy-five participants (85.2%) are aware of this method, and 21 (28%) have already used it. However, considering that games for software engineering

education is not new [2] [6] [21], it is surprising to observe that 13 participants (14.2%) had no knowledge about them.

Regarding the purpose of the games for the participants who have used them, our results show that the knowledge areas of "Project Management", "Software Process", and "Software Requirements" were the most mentioned as topics covered by the use of games. These topics have in common the difficulty in providing meaningful examples relying only in theory or are hard to simulate in educational projects. Oliveira et al. [15] suggests that "traditional approaches usually adapt and simplify problems, thus, reducing their relevance" and that "Problems are also usually linked to prefabricated solutions that do not help them to develop their own ideas to tackle problems".

In general, the participants believed the use of games in software engineering education has been well received by their students. The most recurrent benefits noted by the professors were: higher students interest (7), higher engagement (4), higher motivation (2) and higher content assimilation (2). This result indicates that the students liked the new approach when used and it was clearly noted by their professors. It seems that the common explanatory teaching methodology tends to generate monotony and disinterest over time, and new and different approaches, when used right and in the appropriate moment, renews the students' attention and interaction with the subject, leading to a better content assimilation.

There was no consensus regarding the difficulties in applying the method. We identified that some of these issues could be overcome with more experience by the professors, like "Extra workload for professor", "Games need a specific learn rhythm" and "Implementation in classroom". Other issues seem to be related to chosen games: "Language", "Game comprehension", and "Games are old and outdated". Again, statements that leads to the problem of choosing a game to apply the methodology.

B. *Do professors use gamification in education?*

Our study shows that a substantial number of participants have already used game elements to support software engineering Education. Out of the 75 professors that did knew this methodology, 19 (25.3%) already used in classes. Additionally, considering that only 12 participants have used both approaches, this number is expressive, since gamification is much more recent than the use of serious games in software engineering education [19].

The most popular game elements among the participants were: Points (15), Challenges (13), Quizzes (11), Badges (7), Milestones (7), Levels (7) and Contests (7) were expected due to Souza et al. (2018) study. Some participants even explained how those elements were used:

"In my disciplines, for example, I give the opportunity to the students to exchange quizzes points (fast exercises lists) for deadline extension days (in a limited way, of course)"

The participants also perceived a positive reception of the usage of this methodology. Almost 80% of the ratings were 4 or 5, where 5 was the maximum representing "Great acceptance, no problems" by the students. Rate 4 represented 57.9% of the responses. In this context we had only one occurrence of rate 0 and one of rate 1. One of these bad ratings were related to bad application of the methodology. For instance, the participant who rated 1 stated:

"I had difficulties from not having an integrated environment with Moodle (students and professors platform) (...). We ended having the elements described in classrooms manually, what wasn't attractive"

The most recurrent benefits noticed from the use of game elements was "higher student interest" (8). Regarding the difficulties, the most pointed out was the "Difficulty in measuring performance improvements" (3 occurrences).

C. *What are the reasons for not using games and gamification in software engineering education?*

There is some consensus in the responses for not adopting serious games and gamification in software engineering education. The occurrences were: lack of knowledge about the approaches (19), lack of knowledge about fitting games (14), lack of time (12), don't believe in the approaches (6), lack of interest (5), lack of material (4), lack of resources (2) and difficulty in correct use by students (1). However, further investigation is required to understand specific motivations for different professors and understand possible correlations.

One particular case that came to our attention, for instance, was that all participants under 30 years old are aware of the use of games for teaching software engineering. However, when we consider the years of experience in teaching software engineering, only 13% of the participants with least experience (0 to 5 years) have used games in classroom, while this percentage is 29% and 25% for the groups with 5 to 15, and more than fifteen years of experience, respectively.

One possible cause is that less experienced professors may not have time, space or confidence to use these approaches. In their responses, the reasons they alleged for not using games in classroom were most related to the lack of better understanding or the approach, not finding the adequate games that fit in their classrooms, and the investment (time) required to adapt their classes. Other possible cause is that it is time and effort demanding to find, understand and adapt educational software engineering games for the use in classroom.

However, the responses for question Q18 shows that there is a positive tendency toward using these game-related approaches in the future. Even though there are many scientific papers proposing and evaluating game related approaches for software engineering education, it seems that there is still a problem in disseminating these approaches among professors and providing instructors with relevant resources to facilitate the adoption of these tools. Therefore, it would be relevant for the community to strive for centralized repositories of information about game-related methods for software engineering education.

VI. THREATS TO VALIDITY

In this section, we document potential threats to the study validity and discuss some bias that may have affected the study results. We also explain our actions to mitigate them.

Results: The results presented in the study reflect our interpretation of the data collected from the answers of the

questionnaire. However, the questions were objective enough to enable readers to derive their own interpretations. In addition, there may be several other important issues in the data collected, not yet discovered or reported by us.

Population Sample: Considering the specific population of software engineering professors in Brazil, we believe that our procedure for identifying and inviting relevant participants retrieved a considerable number of candidates for the study (295). However, even considering the participation rate achieved in this study (30.9%), our results may not reflect the opinion of the general community of software engineering educators. Another limitation is that our sample was limited to the software engineering professionals of one country, therefore, it is not possible to generalize the results to the global community of software engineering professors. However, as an initial study we opted to limit our research to one country as we could more easily filter the renowned superior education institutions and their respective software engineering professors. We believe that the research community can help to expand our results by replicating this survey in different countries.

Questionnaire: The validity of the questionnaire may be threatened by ambiguous questions that may compromise the answers of the participants. However, we executed a pilot study with 10 software engineering educators to identify possible problems related to ambiguity, missing response options, and lack of clarity in our questions. Additionally, the questionnaire was created by three researchers, two having experience in software engineering education and research.

VII. Related Work

This section discusses other studies that are somehow related to the present research described in this paper. We consider related works, studies that investigate the state-of-the-practice in the adoption of Game-based Learning and Gamification in software engineering education. We found one study specifically surveying software engineering instructors regarding the use of games in software engineering education [4]. Most of the related work we found are focused on surveying the literature in order to map the existent game-related approaches in software engineering education [2] [6] [21].

Albayrak [4] investigates the factors that influence the instructor's acceptance of the utilization of games (whether serious or not) in undergraduate software engineering education. In this study, a survey was performed with 30 faculty members teaching software engineering in Turkey. Their results show that "the number of hours per week the instructor plays game", "instructor's experience in using games for educational purposes in general", and "instructor's experience in designing games" have significant impact on the instructor's decision to use games in software engineering education. Our results provide a different perspective, aiming on why instructors do not use games in software engineering education. However, the results are convergent in relation to the fact that professors experience and knowledge on the use of games for teaching software engineering is a key factor for using them or not.

With respect to the use of game-related approaches in software engineering education, our results are in accordance with the results of the secondary studies of Souza et al. [21] and Cautifield et al. [6]. Souza et al. [21] mapped game-related approaches to the knowledge areas of the curricular guidelines of ACM/IEEE [1] and found that the knowledge areas "Software Process" (which also includes project management) and "Software Requirements" are the most covered by those approaches. Cautifield et al. [6] mapped 36 studies on the use of games and simulations for software engineering education to SWEBOK areas. Their results show that these games and simulations have been mostly used to cover software engineering management and development processes areas.

Regarding gamification, Alhammad and Moreno [2] identified that the most positively affected aspect was student engagement. In our results, the students' interest was the most positive benefit observed by our participants who used gamification in their classrooms. The authors also observed that points and leaderboards are the most frequently adopted gaming elements; the same applies to challenges and feedback mechanics. Similarly, in our results we observed points and challenges as the most recurrent game elements.

VIII. Conclusion

This paper presented the results of a survey with software engineering professors to investigate the adoption of Game-based Learning and Gamification in software engineering education. We invited 285 professors from 100 renowned higher education institutions in our country to participate in the study. A total of 88 participants answered a questionnaire with 18 questions. The questions were related to three research questions: (RQ1) Do professors use Game-based Learning to support software engineering education? (RQ2) Do professors use Gamification to support software engineering education? (RQ3) What are the reasons for not adopting serious games and gamification in software engineering education?

The results show that both approaches are known by most of the participants, however only a fraction of them have already applied these approaches. Serious games have been most used to cover "Software Process", "Project Management" and "Software Requirements" knowledge area, while "Software Architecture", "Software Maintenance" and "Software Modelling" are the least covered topics. In the case of Gamification, the most used game elements are Points, Quizzes, and Challenges. The main reasons for not using these approaches are related to lack of knowledge, lack of information about relevant games for teaching software engineering, and the lack of time to plan and include these approaches in the classroom. Finally, there is a positive tendency toward the future adoption of these game-related approaches by the participants.

A future work is the creation of a web platform for maintaining and promoting serious games for software engineering, related materials, and resources to support the adoption of game-related approaches in software engineering education. This is intended to facilitate educators interested in the adoption of such approaches.

Acknowledgments

This work was partially supported by CAPES, CNPq (grant 424340/2016-0), and FAPEMIG (grant PPM-00651-17).

REFERENCES

[1] IEEE & ACM JTFCC, "Software Engineering 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering," in IEEE & ACM;The Joint Task Force on Computing Curricula, November 23 2015.

[2] 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.

[3] B. S. Akpolat and W. Slany, "Enhancing Software Engineering Student Team Engagement in a High-Intensity Extreme Programming Dourse using Gamification", IEEE Conference on Software Engineering Education and Training (CSEE&T), 2014

[4] O. Albayrak, "Instructor's Acceptance of Games Utilization in Undergraduate Software Engineering Education: A Pilot Study in Turkey". IEEE/ACM 4th International Workshop on Games and Software Engineering (GAS), 2015.

[5] A. Baker, E. O. Navarro and A. v. d. Hoek, "Problems and Programmers: an Educational Software Engineering Card Game". In 25th International Conference on Software Engineering (ICSE), pp. 614–619., 2003.

[6] C. Caulfield, J. C. Xia, D. Veal and S. P. Maj, "A Systematic Survey of Games Used for Software Engineering Education," Modern Applied Science, 5(6), p. 28, 2011.

[7] S. Deterding and D. Dixon, "Gamification: Using Game Design Elements in Non-Gaming Dontexts". In Extended Abstracts on Human Factors in Computing Systems (CHI). 2011.

[8] S. Easterbrook, J. Singer, M. A. Storey and D. Damian, "Selecting Empirical Methods for Software Engineering Research". Guide to advanced empirical software engineering. Springer-Verlag. pp.285-311, 2008.

[9] E. Fernandes, J. Oliveira, and E. Figueiredo. "Investigating How Features of Online Learning Support Software Process Education". In proceedings of the 46th Frontiers in Education Conference (FIE). Erie, 2016.

[10] E. Figueiredo, J. Pereira, L. Garcia, and L. Lourdes. "On the Evaluation of an Open Software Engineering Course". In proceedings of the 44th Annual Frontiers in Education Conference (FIE). Madrid, Spain, 2014.

[11] Lattes Platform. Available at http://lattes.cnpq.br/. Last access in 20/12/2017.

[12] T. Lima, B. Campos, R. Santos and C. Werner, "UbiRE: A Game for Teaching Requirements in the Context of Ubiquitous Systems". In XXXVIII Conferencia Latinoamericana en Informatica (CLEI). Medellin, Colombia. 2012.

[13] M. R. Marques, A. Quispe and F. S. Ochoa, "A Systematic Mapping Study on Practical Approaches to Teaching Software Engineering". In *Frontiers in Education Conference (FIE)*, 2014.

[14] E. Navarro and A. Van Der Hoek, "Multi-site Evaluation of SimSE". In ACM SIGCSE Bulletin. Vol. 41. No. 1, 2009.

[15] C. Oliveira, M. Cintra and F. Mendes Neto, "Learning Risk Management in Software Projects with a Serious Game based on Intelligent Agents and Fuzzy Systems". In 8th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT), 2013.

[16] S. Pfleeger and B. Kitchenham, "Principles of Survey Research: Part 1: Turning Lemons into Lemonade". *SIGSOFT Software. Engineering Notes*, 26, 6 (November 2001), 16-18. 2001.

[17] M. A. Revilla, S. Manzoor, and R. Liu. "Competitive Learning in Informatics: the UVa Online Judge Experience". Olympiads in Informatics 2, 131–148, 2008.

[18] A. Santos, M. Souza, M. Dayrell, and E. Figueiredo. "Game Elements for Learning Programming: a Mapping Study". In proceedings of the 10th International Conference on Computer Supported Education (CSEDU), Funchal, 2018.

[19] M. Souza, L. Veado, R. Moreira, E. Figueiredo, and H. Xavier. "Games for Learning: Bridging Game-Related Education Methods to Software Engineering Knowledge Areas". In proceedings of the 39th International Conference on Software Engineering (ICSE), Software Engineering Education and Training track, pp. 170-179, Buenos Aires, 2017.

[20] M. Souza, K. Constantino, L. Veado, and E. Figueiredo. "Gamification in Software Engineering Education: An Empirical Study". In proceedings of the 30th International Conference on Software Engineering Education and Training (CSEE&T), pp. 276-284, Savannah, US, 2017.

[21] M. R. A. Souza, 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.

[22] V. Uskov and B. Sekar, "Gamification of Software Engineering Curriculum", Frontiers In Education Conference (FIE), 2014

[23] E. Verdu, L. M. Regueras , M. J. Verdu, J. P. Leal , J. P. Castro , R. Queiros. "A Distributed System for Learning Programming On-Line". Computers & Education, v.58 n.1, p.1-10, January, 2012.

[24] C. G. V. Wangenheim and F. Shull, "To Game or Not to Game?," IEEE Software, vol. 26, no. 2, pp. 92-94, 2009.

[25] S. Wasik, M. Antczak, J. Badura, A. Laskowski, and T. Sternal. "A Survey on Online Judge Systems and Their Applications". ACM Computing Surveys, 51, 1, Article 3, 2018.