Experimental Evaluation of Using Gamification in Software Engineering Education in Software Requirements

Sergio Galvan-Cruz
Centro de Investigacion en Matematicas, Unidad Zacatecas
Zacatecas, Mexico, 98160
sergio.galvan@cimat.mx

and

Mirna Munoz
Centro de Investigacion en Matematicas, Unidad Zacatecas
Zacatecas, Mexico, 98160
mirna.munoz@cimat.mx

and

Jezreel Mejia
Centro de Investigacion en Matematicas, Unidad Zacatecas
Zacatecas, Mexico, 98160
jmejia@cimat.mx

and

Adriana Pena
Universidad de Guadalajara, Centro Universitario de Ciencias Exactas e Ingenierias,
Guadalajara, Mexico, 44430
adriana.ppnegron@academicos.udg.mx

and

David Bonilla Carranza
Universidad de Guadalajara, Centro Universitario de Ciencias Exactas e Ingenierias,
Guadalajara, Mexico, 44430
jose.bcarranza@academicos.udg.mx

Abstract

Education in Software Engineering traditionally focuses on techniques used in the classroom that generate stress and disinterest in the topics taught; therefore, a significant challenge to the use of more effective and motivating teaching methods to achieve student learning. Moreover, the lack of motivating teaching methods results in a lack of perception regarding the importance of artifacts that come together with software, such as documentation. Consequently, students prefer to focus only on development without paying attention to the necessary artifacts, so the maintenance and update of software in later stages are negatively affected. In response to this problem, the need arises to incorporate more dynamic and effective teaching approaches that encourage student participation and commitment. In this sense, Gamification is presented as an innovative solution that has gained increasing interest in Software Engineering and other fields of education. This research aims to provide the results of a case study in which Gamification has been used in teaching Software Engineering for the topic of software requirements within a master's software engineering course.

Keywords: Gamification, Software Engineering, Education, systematic mapping.

1 Introduction

The innovation of teaching methods is still challenging [1], and the file of Software Engineering subjects is not an exception. The teaching methods commonly used in Software Engineering are the traditional ones used in the classroom e.g. a great number of slides containing complicated knowledge and, most of the time, not having practical exercises due to the nature of the knowledge [1, 2, 3]; there are more effective teaching methods are needed since the conventional methods result in students becoming stressed or bored with the topics taught [2, 3]. Besides, they prefer only to get involved in software development rather than their related artifacts, such as documentation or software design[2]. However, not producing the related artifacts complicates software maintenance and updates.

In this context, Gamification provides an innovative solution, and interest in different educational fields, including software engineering, is increasing [4]. Gamification refers to using game design elements in non-game contexts to motivate and guide specific behaviors to achieve objectives [5]. Moreover, it has proven to be a powerful tool to increase student engagement, motivation, and performance by incorporating gamified elements such as point systems, rewards, competitions, and leaderboards [6]. According to Deterding et al. [7], using gamification creates a more engaging and stimulating learning experience. In addition, it allows students to take a more active role in their learning process, effectively involving them in developing skills and knowledge [2, 3, 4, 5].

This research aims to convince students that software development is not just about coding. It highlights the importance of generating relevant artifacts throughout the development of software or system projects, as these artifacts can significantly impact their success.

This research is an extended version of the paper published in CISETC 2023 [8], which identified the state of using gamification to teach Software Engineering topics. Therefore, this paper provides a case study that designs and applies a short experimental evaluation using gamification in a Software Engineering course, specifically the software requirements topic in the Master of Software Engineering degree of CIMAT, Zacatecas- Mexico.

Based on those mentioned above, this research considers that students learn in a more didactic way. Therefore, it is pertinent to implement gamification as part of their teaching. In this way, students increase their involvement and motivation, and therefore, an improvement in their knowledge regarding Software Engineering topics is expected [7].

After the introduction, the structure of this article is as follows: Section 2 provides a background, including an overview of key concepts regarding this research and the results of a systematic mapping; Section 3 presents the case study focused on software requirements; Section 4 provides the results and discussion; and Section 5 provides conclusions and future work.

2 Background

2.1 Key Concepts

- Software Engineering. It was first coined in 1968 at a NATO (North Atlantic Treaty Organization) conference. The topic was related to unreliable products that cost more than expected and were being

delivered late. The Software Engineering term was then focused on the fact that a program development approach alone cannot be applied to develop complex software systems [9, 10]. In 1972, the IEEE Computer Society used the term "Software Engineering" for the first publication on this topic, and some years later (1976), the IEEE Society established a committee to develop software engineering standards [9]. Software Engineering practices involve an intensive job and management of knowledge to achieve a successful product or project [11]. Therefore, Software Engineering aims to provide the resources engineers need to apply the engineering design process to design, develop, test, maintain, and evaluate software [12]. According to the SWEBOK [13], which is the guide to the software engineering body of knowledge, Software Engineering is composed of fifteen knowledge areas listed: software requirements, software design, software construction, software testing, software maintenance, software configuration management, software engineering management, software engineering process, software engineering models and methods, software quality, software engineering professional practice, software engineering economics, computing foundations, mathematical foundations, and engineering foundations. Software has a crucial role today, and because of this, companies demand software engineers with better skills [5]. Unfortunately, there are many unskilled software developers. The education institutions that produce software professionals are the leading cause of this problem because they are still determining the correct way to teach the fundamental concepts and skills students need to apply in the workforce [14]. Software engineering professors must work hard to cover the expectations that students usually have in the final year. Compared to real software development projects, professors lack several critical capabilities of most project managers and owners [14].

- Gamification. It means incorporating game mechanics and dynamics in non-gaming environments or applications to achieve concentration, loyalty, commitment, and other values common to games [15]. The main objectives of Gamification are to motivate people to achieve personal goals and change their behavior or develop new skills [16]. Based on those two objectives, Gamification is presented as a new strategy to influence and motivate people during the development of activities they are not used to or in those in which they are trying to become a new habit [16]. Gamification comprises a set of game dynamics and game mechanics: (1) game dynamics are the needs and motivations that drive people to act, while game mechanics are defined as the strategies used to satisfy those needs [15] and (2) game mechanics include points, achievement or levels, badges, bonuses, tasks, challenges, points systems, progress bars, rewards, scores, leaderboards, feedback, and unblinking content [15, 6]. Nowadays, gamification is extending the use of games in different environments with not just a ludic purpose; its implementation is increasing in fields such as education, health, marketing, and software engineering [2, 3, 4, 5]. Specifically in educational environments, some of its benefits are: 1) promoting effective student interaction in classroom projects, 2) motivating students to learn more, and 3) learning becomes enjoyable [6].

2.2 Systematic mapping

To perform the systematic mapping process authors followed the one proposed by Peterson et al. [10]. Next, a summary of the five steps performed with their results is presented.

2.2.1 Definition of research question

This step focuses on developing a set of research questions that will address the mapping study according to the research goal [10]. This research aims to identify how gamification has been used to teach software engineering topics. Based on this goal, we defined three research questions: RQ1) Which areas of Software Engineering have applied gamification strategies? RQ2) What mechanisms are implemented to evaluate improvements in using gamification to teach software engineering topics? And RQ3) What are the results of using gamification to teach software engineering topics?

2.2.2 Conduct the review

This step focuses on creating a search string(s) and executing it in the database selected [10]. Based on the research questions, we identify a set of keywords that allow us to build a search string: (gamification OR gamified OR gamifying) AND (environments) AND (software engineering). This search string was executed in three digital databases relevant to Software Engineering. As a result, we obtained 531 studies. Table 1 shows the results obtained by the database.

Table 1: Results of executing the search string by database

Scientific Data Base	Obtained studies
IEEE Xplore	75
Scopus	156
ACM Digital Library	300
Total	531

2.2.3 Screening the articles

This step focuses on defining a set of inclusion and exclusion criteria to exclude articles that do not contribute to the research (Petersen et al., 2008). To perform this screening, we established four inclusion criteria (IC) as filters for the articles implemented in two interactions:

- We applied the IC1. to identify the duplicated articles. As a result, we got 211 articles (see Table 2, second column)
- After removing the duplicated studies, we applied three Inclusion Criteria: IC2- articles written in English and Spanish; IC3- availability of articles; and IC4- articles providing implementation of gamification in one or more Software Engineering areas. After applying these three ICs, we obtained 11 articles (see Table 2, third column).

Table 2: Results of executing the four ICs

Scientifics DataBase	Applying IC1	Applying IC2, IC3 and IC4
IEEE Xplore	21	6
Scopus	33	23
ACM Digital Library	157	3
Total	211	11

2.2.4 Keywording using abstracts

This step focuses on developing a classification scheme considering the selected articles (Petersen et al., 2008). For this research, we considered three elements to classify the 11 papers selected: the distribution of articles in Software Engineering areas, the contribution of articles (techniques, tools, practices, and methods), and the type of research according to the research type proposed by Wieringa et al. [17](validation research, evaluation research, solution proposal, philosophical articles, opinion articles, and experience articles). Next, table 3 provides the main results.

Table 3: Main results

Elements	Summary of results
SE areas distribution	The SE areas less covered are software requirements, software design, software maintenance, software configuration management, software engineering professional practice, and software engineering economics, with zero results. Next, with one paper, the areas of software construction, software testing, and software engineering software. Then, with two papers, the areas of software engineering management and software quality. Finally, the areas most covered with four papers are software engineering models and methods
Paper Contribution	• Techniques, which refers to a way of doing an activity related to implementing gamification in the SE area to teach topics (1 paper). • Tools, which refer to software tools that facilitate the implementation of gamification in the SE area to teach topics (2 papers). • Practices, which refer to proven practices related to implementing or using gamification in the SE area to teach topics (4 papers). • Methods, which refers to the "way of work" defined to implement or use gamification in the SE area to teach topics (4 papers).

Type of research	• Evaluation research, which presents techniques, tools, practices, or methods implemented in practice so that authors present how the artifact was implemented and the consequences of its implementation, focusing on benefits, problems, and the implementation evaluation (4 papers). • Validation research, which presents techniques, tools, practices, or methods investigated that are novel and yet to be implemented. They present only work done in a lab (1 paper). • Solution proposal, which presents a solution for a problem related to using gamification to teach topics in some SE areas. It can be a novel solution or an extension of an existing one. It commonly provides potential benefits and analysis of its applicability (4 papers). • Philosophical, which presents a new way of looking at gamification to teach in the SE area, structuring a taxonomy or a conceptual framework (1 paper). • Opinion, which presents an expression of the opinion of somebody related to whether an existing technique, tool, practice, or method is good or bad or what should be done (1 paper). • Experience, which explains what and how a technique, tool, practice, or method has been done in practice according to the author's experience (0 papers).

2.2.5 Data extraction and mapping process

This step focuses on the elements defined in the previous step. Table 4 provides a summary of the main findings

Table 4: Summary of the main findings

Research Questions	**Obteined results**
RQ1. Which areas of Softfware Engineering have applied gamification strategies?	The most covered area is software engineering models and methods with 36.6% of the papers. This area aims to implement a structure on software engineering to make that a systematic, repeatable, and ultimately more success-oriented activity [9]. Areas having an intervention in gamification with 18.2% are software engineering management and software quality. Software engineering management aims to apply management activities (planning, coordination, measurement, monitoring, control, and reporting) to ensure that software products and software engineering services are delivered efficiently, effectively, and for the benefit of interested parties. Besides, software quality refers to the desirable characteristics of software products, to the extent that a particular software product possesses those characteristics, and to the processes, tools, and techniques used to achieve those characteristics [9]. Areas with an intervention of 9.1% are software construction, software testing, and software engineering processes. Software construction refers to the detailed creation of functional software through a combination of coding, verification, unit testing, integration testing, and debugging [9]. Software Testing consists of the dynamic verification of a program that provides the expected behaviors in a finite set of test cases appropriately selected from the generally infinite execution domain [9]. The software engineering process consists of a set of interrelated activities that transform one or more inputs into outputs while consuming resources to achieve the transformation ([9]. Finally, we have not found a proposal related to Software Requirements. Requirements Management is a critical stage in the software development life cycle, which involves identifying, analyzing, and specifying system requirements. Therefore, it is crucial to ensure the quality and efficiency of the developed software, but it often needs help in terms of student's understanding and application of the concepts.
RQ2. What mechanisms are implemented to evaluate improvements in using gamification	

to teach software engineering topics?	The most implemented mechanisms are practices and methods. These mechanisms have been used to evaluate how gamification impacts and improves the learning process in each area to identify which gamification approaches are most effective and beneficial for students. The results of this research helped us better understand how to optimally implement gamification in the teaching of Software Engineering and improve student's educational experience.
RQ3. What are the results obteined from using gamification to teach software engineering topics?	The results obtained from primary studies show that gamification can improve the learning experience, increase student participation and facilitates the learning process in each area of Software Engineering. In addition to increasing software development teams commitment, collaboration, and motivation. Therefore, the research results show that gamification has been effectively applied in the teaching of Software Engineering, demonstrating improvements in student participation, motivation, collaboration, and performance. Moreover, it has proven valuable for fostering a more engaging and effective learning environment.

3 Methodology

This research uses a case study as a methodology to implement the gamified test for master's degree students because it is an important empirical method for experimentation in the software engineering field [art checklist for SECS]. It is feasible to implement in individuals, groups, or organizations. It is composed of 5 steps according to [art checklist for SECS]:

- Case study design: define the study objectives and plan the case study.
- Preparation for data collection: involves defining the procedures and protocols required to gather the data.
- Collecting evidence: execute the case study with the data collection.
- Analysis of data collected: analyze the collected data.
- Reporting: establish conclusions according to the collected data. Each step performed in the case study reported in this article is next described.

3.1 Case study design

The objective of this case study is to determine whether gamification enhances student understanding and retention of software requirements knowledge. Software requirements are the fundamentals upon which software engineering projects are built and serve as the crucial first step in the software development life cycle.

Some important aspects that provide software requirements are 1) clear definitions of the problem; 2) avoid ambiguities; 3) basis for planning; 4) change control; 5) Acceptance criteria; 6) facilitation of testing; 7) improve documentation; 8) risk identification; 9) increased customer satisfaction[12, 18].

Therefore, this case study will use the Kahoot platform to evaluate the impact of gamification on learning in the context of software engineering requirements. The target population is six students of the Software Engineering master's degree offered at CIMAT Zacatecas unit. The students took the Software Engineering I subject from January to May 2024. Within this subject, software requirements are included. On the other hand, 53 undergraduate students from the University of Guadalajara, Mexico, were surveyed; who previously took the software engineering course in the computer, biomedical and electronic engineering programs.

3.2 Preparation for data collection

To collect data, two authors, who were the professors of the Software Engineering I course, designed 10 question tests regarding the covered topics. The test is composed of five questions related to a practical case aided by an image; the other five questions are general questions about software requirements concepts. The questions for the Kahoot test included interactive elements, also, every question has a multiple-choice.

Then the test was adapted to the Kahoot platform (https://kahoot.it). The authors selected Kahoot because it is a game-based learning platform used as an educational technology tool in schools. Besides, it is used to create engaging learning experiences through quizzes, surveys, and games that are referred to as "Kahoots". Some features of the Kahoot platform are 1) interactive quizzes and games, 2) accessibility, 3) collaborative learning, 4) a friendly user interface, 5) integration with Learning Management Systems (LMS), and 6) data and analytics [19, 20].

Figure 1 shows a response to a question on the Kahoot platform. In this case, the user-selected response is incorrect. The platform is very intuitive and graphic, so you can get feedback quickly. In our study, each question was chosen to have three possible options.

Figure 2 shows a question, its image in the Kahoot platform, and its answers. In this type of question, the students who answered the test were also able to view an image within the question which in this case showed a user story with certain characteristics. This type of question also had three response options.

Figure 1: Shows a question in the Kahoot platform

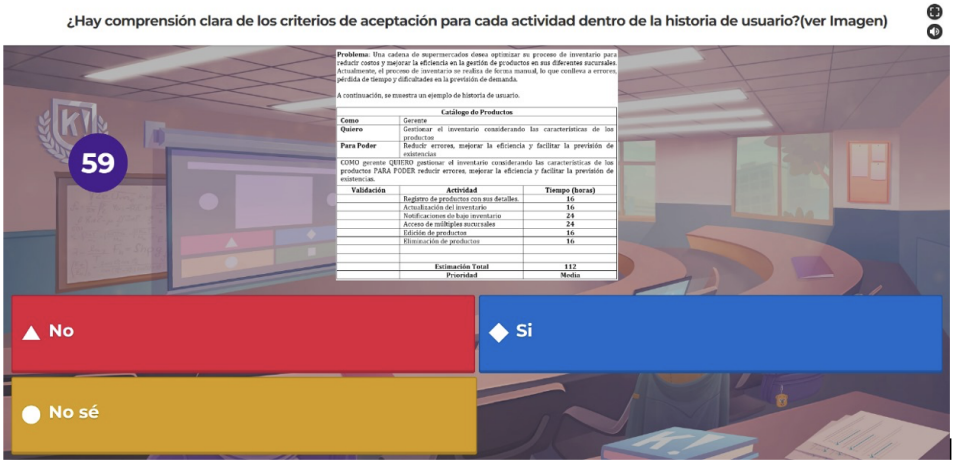

Figure 2: Shows a figure with the image in the Kahoot platform

3.3 Collecting evidence

Professors of the Software Engineering I course sent a message inviting students to answer a brief test related to software requirements. The email contained the link to the 10-question test. Students were also informed that the test had no time limit. Kahoot's features allow us to collect data easily and automatically.

The next steps, analysis of data collected and reporting, will be presented in the next section.

4 Results and Discussion

4.1 Results

This section shows the results obtained from the Kahoot-collected data. Then, we can analyze the professorâs view and the results per question.

Figure 3 shows the results per question; with these results, the professor could implement a strategy for the feedback required by students. Moreover, it facilitates the identification of hard sub-topics.

Pregunta		Tipo	Correcto/incorrecto
1	¿Se prioriza adecuadamente los requisitos en función del valor para el cliente y su impacto en el ...	Quiz	40%
2	¿Hay comprensión clara de los criterios de aceptación para cada actividad dentro de la historia d...	Quiz	60%
3	¿Añadiría actividades de validación a la historia de usuario? (ver Imagen)	Quiz	100%
4	¿Está claro cómo se validará la historia de usuario para verificar el progreso y cumplir los criterio...	Quiz	80%
5	Aspectos como la facilidad de uso y la seguridad, ¿se pudieran agregar en la historia de usuario? ...	Quiz	20%
6	Para definir requisitos, ¿se deben identificar y consultar todas las fuentes de información?	Quiz	100%
7	¿La especificación de requisitos incluye analizar los requisitos y determinar su alcance y factibili...	Quiz	100%
8	¿La especificación de requisitos se debe verificar y validar?	Quiz	100%
9	¿La especificación de requisitos se debe incluir en el manual de usuario?	Quiz	80%
10	La especificación de requisitos cambia constantemente, por lo tanto, no debe ser incluida como ...	Quiz	100%

Figure 3: Results per question obtained by the sample of students.

Figure 4 presents the professor's view about the overall score of every test. As the figure shows, the professor could verify the progress of the respondentâs students, the correct answers and the mistakes. It means, the students were classified by the Kahoot Platform according their scores.

Name	Classification	Correct Answers	Unanswered	Final Score
Student 1	1	100 %	–	9336
Student 2	2	80 %	–	7245
Student 3	3	70 %	–	6850
Student 4	4	70 %	–	6198
Student 5	5	70 %	–	6075
Student 6	1	80 %	–	5976

Figure 4: A score of corrected answers and classification of students.

Additionally, as part of this study, a survey was conducted to gather feedback from participants who completed the gamified study. The main goal of this survey was to understand their perceptions and opinions about the gamified environment they experienced.

The small survey was a component of 6 questions with the following results:

For the first question (effectiveness of Kahoot) with the text: "When I used the Kahoot", the respondents provided diverse feedback, indicating various perceptions about the topic, 14% stated that it was funny, but they did not learn; 71.9% indicated that it was funny and they learned; 12.3% indicated that it was not fun but they learned; finally, 1.8% indicated that it was not fun and they did not have fun. Based on this data, the gamified environment did not demonstrate significant differences compared to a traditional environment. Rather than replacing the traditional approach, the gamified environment complements it, enhancing the overall experience.

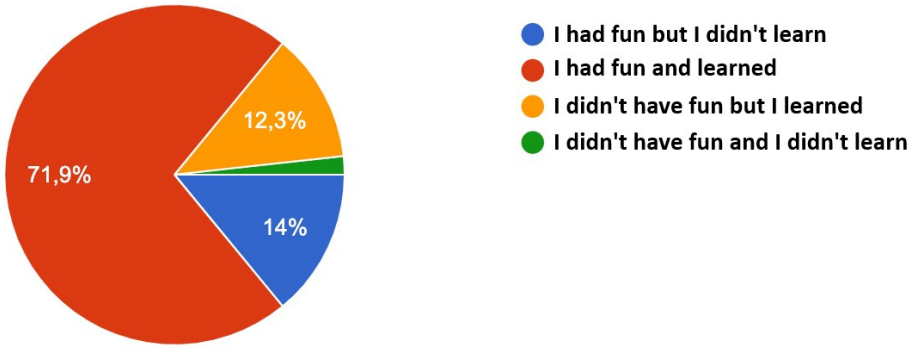

Figure 5: Shows the result of the first question in the feedback survey.

For the second question (knowledge reinforcement) with the text: "Using Kahoot helps to reinforce what I learned from readings or lessons from an instructor", 3.5% of the respondents did not feel that the gamified environment helped them; 26.3%, 49.1%, and 21.1% felt that Kahoot "helps little", "helps" and "helps a lot" respectively. Based on this question, we can conclude that Kahoot is useful for acquiring knowledge.

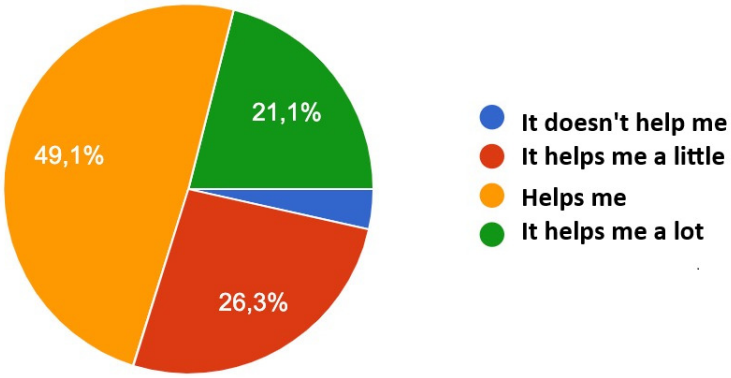

Figure 6: Shows the result of question 2 in the survey feedback.

For the third question (motivation to learn more), the text "Having used Kahoot motivates me to learn more about Software Requirements" mentioned that 8.8% of the respondents said: "It does not motivate;" 31.6% said: "It motivated little myself"; and 49.1% said: "It motivated me"; finally, 10.5% said: âIt motivates me a lotâ. From the above, we can infer that 91.2% of respondents have at least little motivation to use gamification in this case through Kahoot.

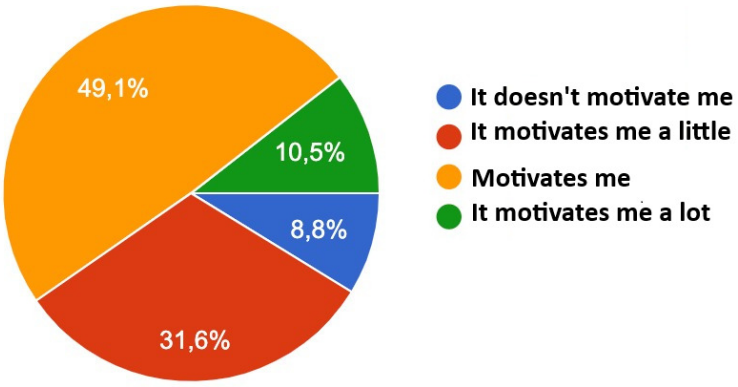

Figure 7: Results of the question number 3 of the feedback survey.

For the results about the fourth question (duration perception) with the text: "In general, the duration of Kahoot seemed to me"; 14% of the respondents indicated that the duration of the gamified environment was "short"; 63.2% felt that it was "regular"; 1.8% felt that it was âlongâ; 19.3% answered that "I did not feel the pass the time"; and finally, 1.8% answered that "exhaustive". Regarding the time, 96.5% answered positively (considering: short, regular and âI did not feel the pass the timeâ), which means the gamified environment has an optimal design according to the respondentâs perception. The above means that the tool used with Kahoot was not tedious and the possibility of its success is high.

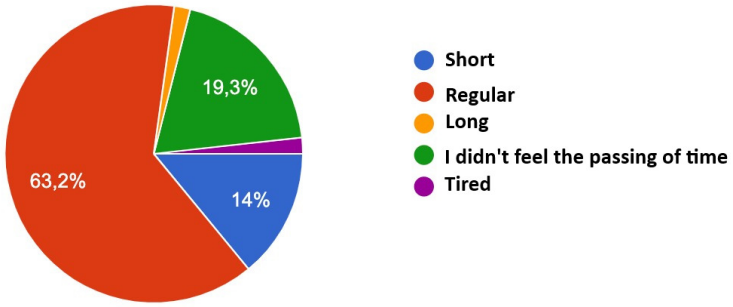

Figure 8: Shows the results of the fourth question in the feedback survey.

For the fifth question (preferred learning method), with the text "I prefer that the instructor explain the theory"; 1.8% of the respondents prefer âexclusively with readingsâ; 93% prefer âreadings mix with gamified environmentsâ; 1.8% prefer âonly gamified environmentsâ; finally, 3.5% of the respondents prefer âother dynamicâ. According to this question, we can conclude that the respondents prefer a combined environment (hybrid environment) for learning some topics, in this case, "software requirements."

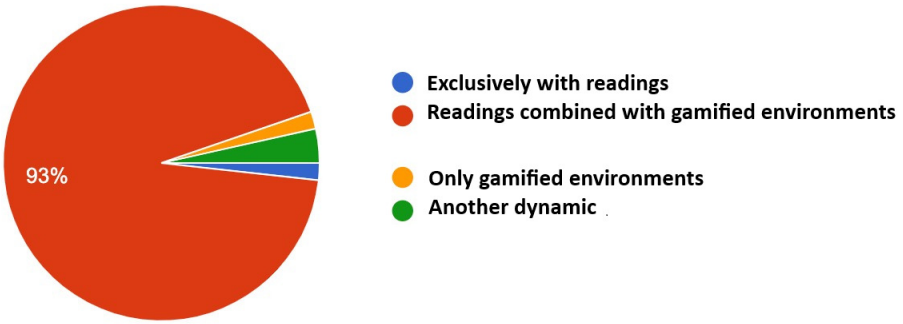

Figure 9: Shows the result of the fifth question in the feedback survey.

Finally, for the sixth question (general perception of gamified environments) with the text "In general, I consider gamified environments to be"; 0% of the respondents said: âunnecessaryâ; 3.5% select the option âwithout importanceâ; 80.7% of the respondents mentioned that this environment is "necessary," while 15.8% noted that it is "essential." In this question, we can conclude that all the respondents feel at least necessary for this type of gamified environment as Kahoot!

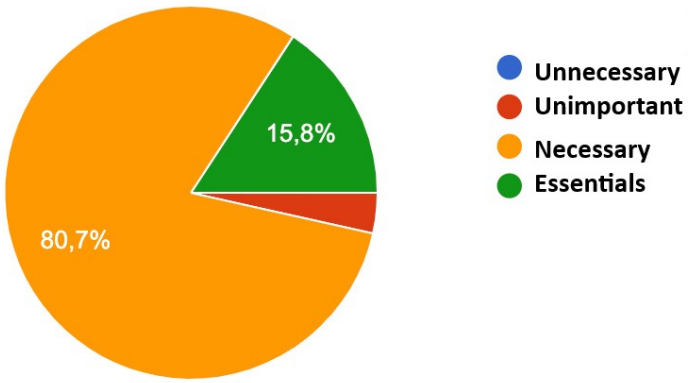

Figure 10: Shows the results of the sixth question in the feedback survey.

4.2 Discussion

The results indicated that the sample of six students who had been trained in software requirements significantly improved in both engagement and performance while reinforcing the topic using the Kahoot platform. The gamified approach was particularly effective in maintaining student interest and promoting active participation during the tests. For example, the immediate feedback provided helped students identify their knowledge gaps and facilitated a deeper understanding of the software engineering requirements concepts [21].

The research findings suggest that gamification can be a valuable tool in enhancing the educational experience in software engineering courses. Also, the professor experience improves using this type of platform because it provides a versatile and user-friendly platform to create engaging and interactive items (such as quizzes, games, etc.) that can easily be integrated into their lesson plans. The Kahoot platform allows professors to apply a quick assessment of students to understand and provide immediate feedback related to a specific topic. Besides, for professors is invaluable for identifying knowledge gaps and adjusting teaching strategies accordingly. On the other hand, for students, the experience is transformed against traditional learning into a fun and competitive experience, significantly boosting motivation and engagement.

5 Conclusions

Implementing Gamification in the teaching of Software Engineering shows excellent potential to improve student's educational experience. Results of the systematic mapping of the studies provided in the background show that software requirements are one of the Software Engineering areas that have less adopted gamification strategies and the mechanisms to evaluate the effectiveness of knowledge taught as part of Software Engineering courses. It is important to highlight that the results obtained so far can be generalized.

Two elements should be highlighted: 1) the need to create gamification strategies that address less explored areas, such as Software Requirements, which can help professors obtain quick feedback on their students' knowledge and understanding, and 2) the necessity for extensive validation to confirm the effectiveness of this type of strategy with a larger sample of students in a real educational environment.

The case study results demonstrated that the use of digital platforms, such as Kahoot, increases engagement and motivation and enhances understanding and retention of key concepts related to specific topics. In this paper, we present the software requirements.

These results also suggest that integrating gamified elements into educational practices can be a powerful strategy for improving learning courses. The feedback survey results indicate that while Kahoot's gamified environment may not significantly outperform traditional methods in all aspects, it is generally perceived as a useful, essential, motivating, and well-designed tool that reinforces knowledge and supports a hybrid learning approach. The respondents unanimously recognize the importance of incorporating gamified elements into educational practices.

In conclusion, gamification can be a valuable tool to improve the teaching of Software Engineering and foster a more stimulating and participatory learning environment. This research is a starting point for designing more effective educational strategies in Software Engineering. This area provides a solid foundation for future research and environments in secondary and higher education institutions. Therefore, future research could explore the long-term effects of gamified testing and its applicability to other areas of software engineering education.

References

[1] G. Martínez Villalobos and J. F. Ríos Herrera, "Gamificación como estrategia de aprendizaje en la formación de estudiantes de ingeniería," *Estudios pedagógicos (Valdivia)*, vol. 45, no. 3, pp. 115–125, 2019.

[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] S. Tonhão, M. Shigenaga, J. Herculani, A. Medeiros, A. Amaral, W. Silva, T. Colanzi, and I. Steinmacher, "Gamification in software engineering education: a tertiary study," in *Proceedings of the XXXVII Brazilian Symposium on Software Engineering*, 2023, pp. 358–367.

[4] R. H. B. Monteiro, M. R. de Almeida Souza, S. R. B. Oliveira, C. dos Santos Portela, and C. E. de Cristo Lobato, "The diversity of gamification evaluation in the software engineering education and industry: Trends, comparisons and gaps," in *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*. IEEE, 2021, pp. 154–164.

[5] M. Muñoz and G. P. Gasca-Hurtado, "Gamificación para atender los desafíos de la enseñanza ingeniería de software en instituciones de educación superior," *Revista Ibérica de Sistemas e Tecnologias de Informação*, no. 49, pp. 5–21, 2023.

[6] M. Muñoz, A. P. Pérez Negrón, J. Mejia, G. P. Gasca-Hurtado, M. C. Gómez-Alvarez, and L. Hernández, "Applying gamification elements to build teams for software development," *IET Software*, vol. 13, no. 2, pp. 99–105, 2019.

[7] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: defining" gamification"," in *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, 2011, pp. 9–15.

[8] W. Avila-Alvarado, M. Muñoz, S. Galvan-Cruz, J. Mejía, and A. Peña, "Exploratory analysis of using gamification in software engineering education." in *CISETC*, 2023, pp. 20–30.

[9] A. Abran, J. W. Moore, P. Bourque, R. Dupuis, and L. L. Tripp, *Guide to the software engineering body of knowledge: 2004 version SWEBOK*. IEEE Computer Society, 2004.

[10] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *12th international conference on evaluation and assessment in software engineering (EASE)*. BCS Learning & Development, 2008.

[11] F. O. Bjørnson and T. Dingsøyr, "Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used," *Information and software technology*, vol. 50, no. 11, pp. 1055–1068, 2008.

[12] R. S. Pressman, *Software engineering: a practitioner's approach*. Palgrave macmillan, 2005.

[13] H. Washizaki, "An overview of the swebok guide."

[14] D. Coppit and J. M. Haddox-Schatz, "Large team projects in software engineering courses," *ACM SIGCSE Bulletin*, vol. 37, no. 1, pp. 137–141, 2005.

[15] O. Borrás-Gené *et al.*, "Introducción a la gamificación o ludificación (en educación)," 2022.

[16] B. Burke, *Gamify: How gamification motivates people to do extraordinary things*. routledge, 2016.

[17] R. Wieringa, N. Maiden, N. Mead, and C. Rolland, "Requirements engineering paper classification and evaluation criteria: a proposal and a discussion," *Requirements engineering*, vol. 11, pp. 102–107, 2006.

[18] I. Sommerville, "Software testing," *Software Engineering*, vol. 8, pp. 537–65, 2010.

[19] C. M. Plump and J. LaRosa, "Using kahoot! in the classroom to create engagement and active learning: A game-based technology solution for elearning novices," *Management Teaching Review*, vol. 2, no. 2, pp. 151–158, 2017.

[20] A. I. Wang, "The wear out effect of a game-based student response system," *Computers & Education*, vol. 82, pp. 217–227, 2015.

[21] N. V. Iaremenko, "Enhancing english language learnersâ motivation through online games," *Information Technologies and Learning Tools*, no. 59, issue. 3, pp. 126–133, 2017.