Exploratory Analysis of Using Gamification in Software Engineering Education

Wendy Avila-Alvarado[1], Mirna Muñoz[2], Sergio Galvan-Cruz[2], Jezreel Mejía[2], Adriana Peña[3]

[1] TecNM Tianguistenco, Carretera Tenango, Santiago - La Marquesa 22, 52650 Santiago Tilapa, Estado de México, Méxicoo
[2] Centro de Investigación en Matemáticas, Calle Lasec y Andador Galileo Galilei, Manzana 3, Lote 7 Quantum Ciudad del Conocimiento, 98160, Zacatecas, México
[3] Universidad de Guadalajara- Centro Universitario de Ciencias Exactas e Ingenierías, Blvd. Gral. Marcelino García Barragán 1421, Olímpica, 44430 Guadalajara, México

Abstract

Software Engineering education faces significant challenges regarding effective and motivating teaching methods to achieve student learning. Most of the techniques used in the classroom are conventional, which can generate stress and disinterest in the topics taught. The lack of motivating methods results in a lack of interest in documentation, so students prefer to focus only on development without paying attention to the necessary documentation. This fact can negatively affect maintaining and updating software in later stages. 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 identify how Gamification has been used in teaching Software Engineering through a systematic mapping study that analyzes the existing primary studies on this topic.

Keywords

Gamification, Software Engineering, Education

1. Introduction

The student learning a Software Engineering subject needs more effective learning methods, so the techniques are conventional. This fact results in students becoming stressed or bored with the topics taught in this subject. Besides, they prefer only to get involved in software development rather than in their documentation. However, this lack of documentation makes software maintenance and updates challenging to perform.

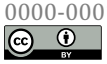
Gamification is presented as an innovative solution that has gained increasing interest in Software Engineering and other fields of education. Gamification refers to using game design elements in non-game contexts to motivate and guide specific behaviours to achieve particular objectives [1]. Gamification in education 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 [2]. Gamification creates a more engaging and stimulating learning experience [3]. In addition, it allows students to take a more active role in their learning process, effectively involving them in developing skills and knowledge.

This research identifies the state of using gamification to teach Software Engineering topics. This need arises to convince students that software development is not only code. The generation

CISETC 2023: International Congress on Education and Technology in Sciences, December 04–06, 2023, Zacatecas, Mexico

wendy_201923012@test.edu.mx (W. Avila-Alvarado); mirna.munoz@cimat.mx (M. Muñoz); jmejia@cimat.mx (J. Mejía); sergio.galvan@cimat.mx (S. Galvan-Cruz); adriana.ppnegron@academicos.udg.mx (A. Peña)

0009-0009-9313-3200 (W. Avila-Alvarado); 0000-0001-8537-2695 (M. Muñoz); 0000-0003-0292-9318 (J. Mejía); 0000-0001-7453-3115 (S. Galvan-Cruz) 0000-0001-6823-2367 (A. Peña).



of relevant artifacts throughout the development of software or system projects can impact its success.

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 software development performance is expected [3]. Therefore, to understand the use of gamification in teaching software engineering topics, we perform a systematic mapping to identify the state related to the obtained results.

After the introduction, the structure of this article is as follows: section 2 provides an overview of key concepts regarding this research; Section 3 shows the systematic mapping defined to perform the analysis; Section 4 presents the results and discussion; and Section 5 provides conclusions and future work.

2. Background

2.1. Software Engineering

In 1958, Jhon Turkey coined "software" [4]. "Software Engineering" was first coined in 1968 at a NATO (North Atlantic Treaty Organization) conference discussing what was then called the "software crisis," which was related to unreliable products that cost more than expected and were being delivered late. Then, the Software Engineering term was focused on the fact that a program development approach alone cannot be applied to develop complex software systems [4, 5]. 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 [4].

Software Engineering practices involve an intensive job and management of knowledge to achieve a successful product or project [6]. Therefore, Software Engineering aims to provide the resources engineers need to apply the engineering design process to design, develop, test, maintain, and evaluate software [7], [8].

According to the SEWBOK [10], 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.

An essential activity in Software Engineering is Project Management (PM) because it involves planning, coordinating, and controlling the activities in IT projects that share an important characteristic: "the projection of ideas and activities into new endeavours." There are many examples of projects with exceeded time and cost ending in unsuccessful terms, even being abandoned before its completion. The objective of PM is to avoid and warn problems as much as possible through plan, organizing, and controlling activities. The process starts before the resources are committed and keeps during all the advances until the delivery or even the end of the project [9].

Project Management (PM) activities in projects before 1900 have legacies from an architectural and industrial culture based on past centuries. The formal management of buildings and military structures arose during the Ming dynasty (1368-1644). During the first years of the 20th century, rapid industrialization as production-line manufacture (driven by Henry Ford) and munitions production (World War 1) showed the importance of tracking and controlling projects efficiently. For the second half of the century (1950-1969), the emergence of computing (hardware and software) encouraged greater importance on the management of the projects to be developed. Finally, in the IT industry, the importance of PM from 1990 to the present has made possible the development of complex and large systems that we use in our daily lives. [9].

The period between the beginning and the end of the project is called the "project life cycle," in this period, three roles are essential: customer, contractor, and project manager. However, the above does not apply to all project types [9]. A project manager is vital to correctly using PM on a project. According to PMBOK [11], the ability to motivate the team and other stakeholders in the project is a skill crucial to the project manager. Even the project manager's organizational leadership implies that others carry out their activities efficiently [11].

Software has a crucial role today, and because of this, companies demand software engineers with better skills. 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 [12]. 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 [12].

2.2. Gamification

Gamification means incorporating game mechanics and dynamics in non-gaming environments or applications to achieve concentration, loyalty, commitment, and other values common to games [13].

The main objectives of Gamification are to motivate people to achieve personal goals and change their behaviour or develop new skills [14]. 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 [14].

Gamification comprises a set of game dynamics and game mechanics:

- Game dynamics are defined as the needs and motivations that drive people to act, while game mechanics are defined as the strategies used to satisfy those needs [13].
- Game mechanics include points, achievement or levels, badges, bonuses, tasks, challenges, points systems, progress bars, rewards, scores, leaderboards, feedback, and unblinking content [13][15].

Gamification is not new; it began at the end of 2010 and the beginning of 2011. Gamification was first used for applications in areas such as 1) health in applications aimed at performing physical activities and 2) social interaction and learning applications, such as Foursquare or Duolingo, where the implementation of points and table positions is common as motivating elements.

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. Specifically in educational environments, some of its benefits are [15]: 1) promoting effective student interaction in classroom projects, 2) motivating students to learn more, and 3) learning becomes enjoyable.

2.3. Related work

This section identifies previous works related to Gamification and Software Engineering research. Then, this section presents five related articles performed within the last six years.

Erika et al., in Gamification for software process improvement software: an exploratory analysis [16], performed an SLR focused on incorporating gamification elements into the Software Engineering (SE) field, especially in processes. They found 24 primary studies. The main results highlighted by this article are: 1) areas in SE in which gamification strategies have been implemented are software development, software engineering (those containing more than one area), software testing, programming, software design and software quality; 2) around 70% of studies use tools to support processes to implement gamification; 3) most of the primary studies

are related to the educational field; 4) most of the gamification elements in processes are points, levels and position tables; and 5) 20 of 24 primary studies have positive effects.

Martinez-Villalobos and Rios-Herrera, in Gamification as a learning strategy in the training of engineering students [17], performed a research project related to the use of gamification and pedagogical strategies to train engineering students of programs related to civil engineering, systems, electronics, mechanical and industrial. The project consisted of using a video game with 40 students to be implemented to teach the subject of linear algebra. The methodological and design criteria that Gamification most impacts are motivation (93%), utility (91%), and evaluation (91%).

Muñoz and Gasca-Hurtado, in Gamification for Addressing the Challenges of Teaching International Software Engineering Standards in Higher Education Institutions [18], addressed the challenges faced in developing the skills required to use software engineering standards (such as commitment, training, understanding and conscientization). And how gamification elements can contribute to education on this critical topic to develop quality software products.

Abraham and Moreno, in Gamification in Software Engineering Education: A systematic mapping [19], identified a relationship between gamification and Software Engineering in aspects such as 1) gamification helps to improve student engagement, 2) sharing of knowledge, and 3) encouraging the best practices. After the verification of 127 studies retrieved, the authors obtained 40 papers marked as relevant, and the Gamification remarks some software engineering processes as the following: construction, improvement, testing, maintenance, risk, design, requirements, and configuration.

Tonhão et al., in Gamification in Software Engineering Education: a Tertiary Study [20], mentioned that in the educational scope, almost half about university courses, and a similar quantity with learning experiences. Currently, the importance of SE keeps growing in the software industry; therefore, gamification techniques are implemented for training purposes. The studies focused on structural gamification, i.e., it implies game elements to modify the learning environment without altering the content. The results show the potential of gamification to improve the engagement and motivation of the students across the knowledge process.

Monteiro, in The Diversity of gamification evaluation in the software engineering education and industry: Trends, comparisons and gaps [21], focuses on the evaluation based on the user experience and perceptions as a strategy for implementing gamification and evaluating the results and the effects of gamification on its users and their context.

Even when the research works present an analysis of gamification and SE, they need to analyze the coverage of gamification strategies by areas of SE, which is the main topic of this paper.

3. Systematic mapping

The systematic mapping process was proposed by Peterson et al. [22]. The process suggests five steps: 1) definition of the research question, 2) conduct of the review, 3) screening of the articles, 4) keywording using abstracts, and 5) data extraction and mapping process. The adaptation of each step to this research is next described.

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 [22]. 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. *Conduct the review:* this step focuses on creating a search string(s) and executing it in the database selected [22]. 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

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 [22]. To perform this screening, we established four inclusion criteria (IC) as filters of the articles implemented in two interactions:
    a. We applied the IC1. to identify the duplicated articles. As a result, we got 211 articles (see Table 2, second column)
    b. 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). The list of the 11 articles is provided in Annex A.

Table 2
Results of executing the four ICs

Scientifics Data Base	Applying IC1	Applying IC2, IC3 and IC4
IEEE Xplore	21	6
Scopus	33	2
ACM Digital Library	157	3
Total	211	11

4. *Keywording using abstracts:* this step focuses on developing a classification scheme considering the selected articles [22]. For this research, we considered three elements: 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. [23] (validation research, evaluation research, solution proposal, philosophical articles, opinion articles, and experience articles).
5. *Data extraction and mapping process:* this step focuses on the elements defined in the previous step. We will present the results of the systematic mapping in section 4.

4. Results and Discussion

4.1 Distribution of articles by Software Engineering areas

As mentioned before, according to SWEBOK, Software Engineering (SE) has 15 knowledge areas. Then, we focused on 12 of 15 areas to classify the articles and identify which SE areas are concentrated in the selected articles. We do not consider computing foundations, mathematical foundations, and engineering foundations. Table 3 shows the papers' distribution. As the table shows, software engineering models and methods are the SE area most focused on, with four articles, followed by software engineering management and software quality, with two articles.

Finally, we found one article on software construction, design, and engineering processes. We need to find evidence of articles addressing the use of Gamification in the rest of the SE areas.

Table 3
Distribution of articles by SE areas

SE areas	# of articles	frequency
Software requirements	0	0.0%
Software design	0	0.0%
Software construction	1	9.1%
Software testing	1	9.1%
Software maintenance	0	0.0%
Software configuration management	0	0.0%
Software engineering management	2	18.2%
Software engineering process	1	9.1%
Software engineering models and methods	4	36.3%
Software quality	2	18.2%
Software engineering professional practice	0	0.0%
Software engineering economics	0	0.0%

4.2 Type of contribution of the articles

For this research, we classify the contribution of the most common elements found in a SE area, such as techniques, tools, practices, and methods. Each term is briefly described. Besides, Table 4 presents the classification of articles.

- *Techniques:* refers to a way of doing an activity related to implementing gamification in the SE area to teach topics.
- *Tools:* refer to software tools that facilitate the implementation of gamification in the SE area to teach topics.
- *Practices:* refer to proven practices related to implementing or using gamification in the SE area to teach topics.
- *Methods:* it refers to the "way of work" defined to implement or use gamification in the SE area to teach topics.

Table 4
Type of contribution of articles by SE areas

SE areas	Techniques	Tools	Practices	Methods	Total
Software requirements	0	0	0	0	0
Software design	0	0	0	0	0
Software construction	0	[E1]	0	0	1
Software testing	0	0	[E2]	0	1
Software maintenance	0	0	0	0	0
Software configuration management	0	0	0	0	0
Software engineering management	[E3]	0	[E4]	0	2
Software engineering process	0	0	[E5]	0	1
Software engineering models and methods	0	[E6]	0	[E7], [E8], [E9]	4
Software quality	0	0	[E10]	[E11]	2

Software engineering professional practice	0	0	0	0	0
Software engineering economics	0	0	0	0	0
Total	1	2	4	4	11

As the table shows, most articles focus on proven practices and methods, with four articles each, followed by tools with two articles. Finally, we only find one paper related to techniques. This could be because of the need for strategies that allow them to follow a structured way to implement gamification in SE areas.

4.3 Research type of articles

For this research, we classify the research type according to the classification of research type proposed by Wieringa et al. [23], adapting this classification to the context of our research. Each term is briefly described, and Table 5 presents the classification of the articles.

- *Evaluation research (ER):* the paper 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.
- *Validation research (VR):* the paper presents techniques, tools, practices, or methods investigated that are novel and yet to be implemented. They present only work done in a lab.
- *Solution proposal (SP):* the paper 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.
- *A philosophical paper (PP):* the paper presents a new way of looking at gamification to teach in the SE area, structuring a taxonomy or a conceptual framework.
- *Opinion paper (OP):* the paper 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.
- *Experience paper (EP):* the paper explains what and how a technique, tool, practice, or method has been done in practice according to the author's experience.

Table 5
Type of research articles by SE areas

SE areas	ER	VR	SP	PP	OP	EP	Total
Software requirements	0	0	0	0	0	0	0
Software design	0	0	0	0	0	0	0
Software construction	0	0	[E1]	0	0	0	1
Software testing	[E2]	0	0	0	0	0	1
Software maintenance	0	0	0	0	0	0	0
Software configuration management	0	0	0	0	0	0	0
Software engineering management	0	[E3]	0	0	[E4]	0	2
Software engineering process	[E5]	0	0	0	0	0	1
Software engineering models and methods	0	0	[E7], [E8], [E9]	[E6]	0	0	4
Software quality	[E10], [E11]	0	0	0	0	0	2

Software engineering professional practice	0	0	0	0	0	0	0
Software engineering economics	0	0	0	0	0	0	0
Total	4	1	4	1	1	0	11

As the table shows, most papers are the type of evaluation research (ER) and solution proposal (SP), with four papers each. Besides, validation research (VR), philosophical paper (PP) and opinion paper (OP) with one paper each.

The mapping study results focused on the distribution of articles by SE area, type of contribution, and type of research are provided in Figure 1.

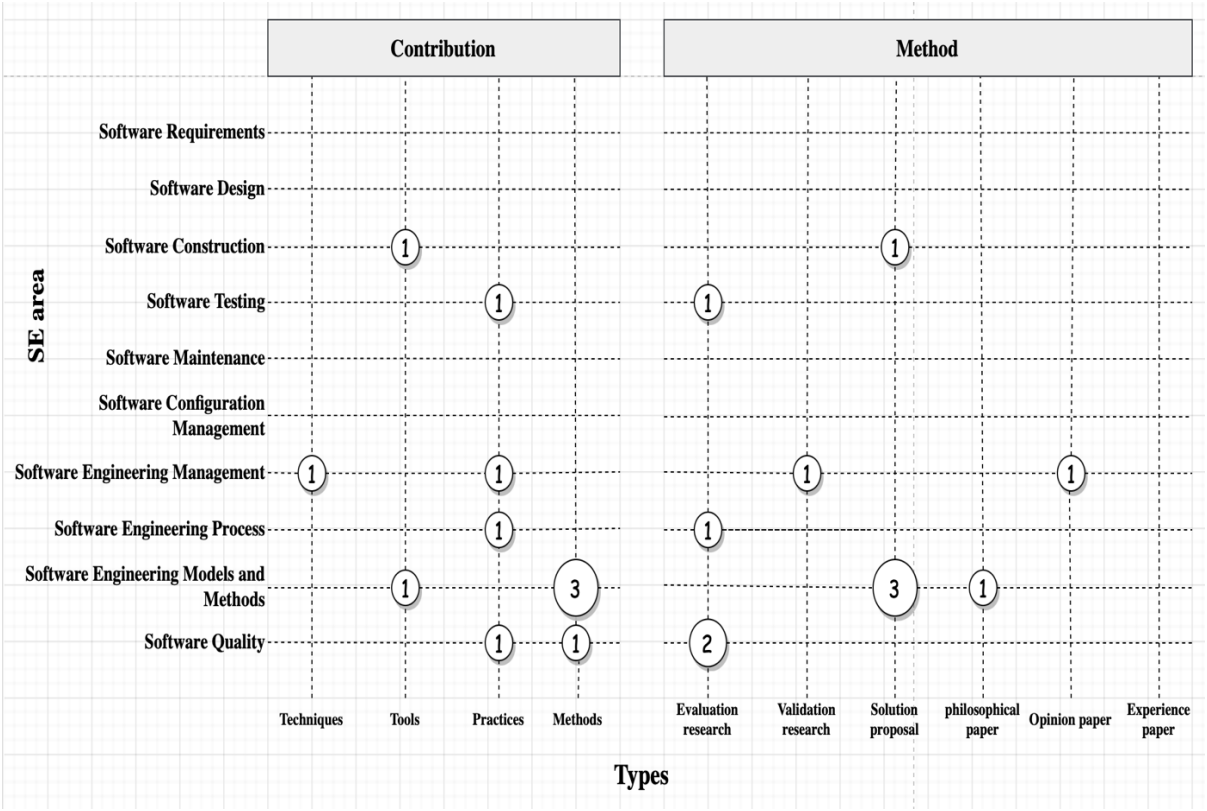


Figure 1: Mapping study results focused on the distribution of SE area, type of contribution, and type of research

4.4 Discussion

This section will be focused on answering the RQs defined to address the research.

RQ1. Which areas of Software Engineering have applied gamification strategies?

To answer this question, we classified the gamification strategies to implement gamification for software process improvement in the SE areas. The areas with the most significant intervention in gamification are software construction, software testing, software engineering management, software engineering processes, software engineering models and methods, and software quality (see Table 3).

The area that stands out mainly, with 36.3%, is the area of software engineering models and methods (E6, E7, E8, E9). This area aims to implement a structure on software engineering to make that a systematic, repeatable, and ultimately more success-oriented activity [4].

Other areas that have had an intervention in gamification with 18.2% are software engineering management (E3, E4) and software quality (E10, E11). 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 [4].

Finally, the areas with an intervention of 9.1% are software construction (E1), software testing (E2), and software engineering processes (E5). Software construction refers to the detailed creation of functional software through a combination of coding, verification, unit testing, integration testing, and debugging [4]. Software Testing consists of the dynamic verification of a program that provides the expected behaviours in a finite set of test cases appropriately selected from the generally infinite execution domain [4]. 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 [4].

It is of particular interest to highlight that 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 students' understanding and application of the concepts.

RQ2. What mechanisms are implemented to evaluate improvements in using gamification to teach software engineering topics?

Table 4 shows the classified types of mechanisms most used to improve software engineering processes in each area. The most implemented mechanisms are practices (E2, E4, E5, E10) and methods (E7, E8, E9, E11).

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 students' educational experience.

RQ3. What are the results obtained from using gamification to teach software engineering topics?

The results obtained from primary studies show us that gamification can improve the learning experience and increase student participation (E3). They also demonstrate that gamification facilitates the learning process in each area of Software Engineering (E1, E5). In addition to increasing software development teams' commitment, collaboration, and motivation (E4). 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 (E10). Gamification has proven valuable for fostering a more engaging and effective learning environment (E7, E8, E11).

5. Conclusion

Implementing Gamification in the teaching of Software Engineering shows excellent potential to improve students' educational experience. Through the systematic mapping of the studies, it was possible to identify the areas of Software Engineering that have most adopted gamification strategies and the mechanisms used to evaluate their effectiveness management.

On the one side, the related work provides the research interest in using gamification to teach engineering areas. Besides, they give an overview of the usefulness of gamification of these topics.

On the other side, according to the results of the systematic mapping, gamification elements, such as point systems, leaderboards, and rewards, are intended to encourage participation and friendly competition among students.

It is important to highlight that the results obtained so far show a positive impact of gamification on the Software Engineering learning process. However, two elements should be highlighted: 1) the need to create gamification strategies covering lees-covered areas such as

Software Requirements, and 2) extensive validation is required to confirm the effectiveness of this strategy in the real educational environment.

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.

References

[1] E. Bonilla-Rivas, Estrategia para la capacitación en estándares de desarrollo de software basado en un juego serio: aplicación al estándar ISO/IEC 29110, Master's thesis, Centro de Investigación en Matemáticas A.C., Zacatecas, México, 2021.

[2] Muñoz, Mirna; Pérez Negrón, Adriana Peña; Mejía, Jezreel; Gasca-Hurtado, Gloria Piedad; Gómez-Álvarez, María Clara; Hernández, Luis. Applying gamification elements to build teams for software development, IET Software, April 2019, 13, (2), p. 99-105, DOI: 10.1049/iet-sen.2018.5088. IET Digital Library, https://digital-library.theiet.org/content/journals/10.1049/iet-sen.2018.5088

[3] Deterding, S., Dixon, D. A., Khaled, R., & Nacke, L. E. (2011). From game design elements to gamefulness. https://doi.org/10.1145/2181037.2181040.

[4] IEEE Computer Society Professional Practices Committee. (2004). SWEBOK: Guide to the Software Engineering Body of Knowledge, 2004 version. IEEE Computer Society.

[5] Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). Systematic mapping studies in software engineering. En BCS Learning & Development. British Computer Society. https://doi.org/10.14236/ewic/ease2008.8

[6] Bjørnson, F. O., & Dingsøyr, T. (2008). Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. Information and software technology, 50(11), 1055-1068

[7] J Pressman, Roger S (2009). Software Engineering: A Practitioner's Approach (7th ed.). Boston, Mass: McGraw-Hill. ISBN 978-0-07-337597-7.

[8] Sommerville, Ian (2010) [2010]. Software Engineering (9th ed.). Harlow, England: Pearson Education. ISBN 978-0-13-703515-1

[9] Lock, D. (2020). Project management. Routledge.

[10] P. Bourque and R.E. Fairley, eds., Guide to the Software Engineering Body of Knowledge, Version 3.0, IEEE Computer Society, 2014; www.swebok.org.

[11] Guide, A. (2001). Project management body of knowledge (pmbok® guide). In Project Management Institute (Vol. 11, pp. 7–8).

[12] Coppit, D., & Haddox-Schatz, J. M. (2005). Large team projects in software engineering courses. ACM SIGCSE Bulletin, 37(1), 137–141.

[13] Borrás-Gené, O. (2022). Introducción a la gamificación o ludificación (en educación). Madrid. Servicio de Publicaciones de la Universidad Rey Juan Carlos.

[14] Burke, B. (2014). Gamify How Gamification Motivates People to do Extraordinary Things. Bibliomotion, Inc.

[15] Mirna Muñoz, Luis Hernández, Jezreel Mejia, Gloria Piedad Gasca-Hurtado, and María Clara Gómez-Alvarez State of the use of gamification elements in software development teams. Springer International Publishing AG 2017 J. Stolfa et al. (Eds.): EuroSPI 2017, CCIS 748, pp. 249–258, 2017. DOI: 10.1007/978-3-319-64218-5_20

[16] Arceo, E. E. B., Vera, R. A. A., Mendoza, J. C. D., & Pech, J. P. U. (2019). Gamificación para la mejora de procesos en ingeniería de software: Un estudio exploratorio. ReCIBE. *Revista electrónica de Computación, Informática, Biomédica y Electrónica*, 8(1), 1-19.

[17] Martínez Villalobos, G., & Ríos Herrera, J. F. (2019). Gamificación como estrategia de aprendizaje en la formación de estudiantes de Ingeniería. Estudios pedagógicos (Valdivia), 45(3), 115-125.

[18] Muñoz, M., & Gasca-Hurtado, G. P. (2023). Gamificación para atender los desafíos de la enseñanza Ingeniería de Software en instituciones de educación superior. *Revista Ibérica de Sistemas e Tecnologias de Informação*, (49), 5-21.
[19] Alhammad, M. M., & Moreno, A. M. (2018). Gamification in software engineering education: A systematic mapping. Journal of Systems and Software, 141, 131-150
[20] Tonhão, S., Shigenaga, M., Herculani, J., Medeiros, A., Amaral, A., Silva, W., ... & Steinmacher, I. (2023, September). Gamification in Software Engineering Education: a Tertiary Study. In Proceedings of the XXXVII Brazilian Symposium on Software Engineering (pp. 358-367).
[21] Monteiro, R. H. B., de Almeida Souza, M. R., Oliveira, S. R. B., dos Santos Portela, C., & de Cristo Lobato, C. E. (2021, May). 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) (pp. 154-164). IEEE.
[22] Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). Systematic mapping studies in software engineering. In BCS Learning & Development. British Computer Society. https://doi.org/10.14236/ewic/ease2008.8
[23] Wieringa, R., Maiden, N. A. M., Mead, N. R. & Rolland, C. (2006), 'Requirements engineering paper classification and evaluation criteria: a proposal and a discussion', Requir. Eng. 11(1), 102–107

Annex A. Primary studies

(E1) Gasca-Hurtado, G. P., Negrón, A. P. P., Hincapi, J. A., & Zepeda, V. V. (2021c). Gamification of an Educational Environment in Software Engineering: Case Study for Digital Accessibility of people with disabilities. Iberoamericana de Tecnologias del Aprendizaje, Vol. 16, No. 4, November 2021, 1. https://doi.org/10.1109/rita.2021.3137372
(E2) Moser, G., Vallon, R., Bernhart, M., & Grechenig, T. (2021). Teaching Software Quality Assurance with Gamification and Continuous Feedback Techniques. https://doi.org/10.1109/educon46332.2021.9453921
(E3) Soares, E. M., & Oliveira, S. R. B. (2022). An Evaluation of the Participants' Perception of a Gamification to solve Problems of Software Process Improvement in the Educational Context. En 2022 IEEE Frontiers in Education Conference (FIE). https://doi.org/10.1109/fie56618.2022.9962464
(E4) Hernandez, L., Muñoz, M., Mejia, J., & Negrón, A. P. P. (2016b). Gamification in software engineering teamworks: A systematic literature review. https://doi.org/10.1109/cimps.2016.7802799
(E5) Barreto, C. E., & Franca, C. L. (2021). Gamification in Software Engineering: A literature Review. https://doi.org/10.1109/chase52884.2021.00020
(E6) Trinidad-Fernández, M., Calderón, A., & Ruiz, M. (2021). GoRace: A Multi-Context and Narrative-Based Gamification Suite to Overcome Gamification Technological Challenges. IEEE Access, 9, 65882-65905. https://doi.org/10.1109/access.2021.3076291
(E7) Blanco, R., Trinidad, M., Suárez-Cabal, M. J., Calderón, A., Ruiz, M., & Tuya, J. (2023). Can gamification help in software testing education? Findings from an empirical study. Journal of Systems and Software, 200, 111647. https://doi.org/10.1016/j.jss.2023.111647
(E8) Grey, S., & Gordon, N. (2023). Motivating students to learn how to write code using a gamified programming tutor. Education Sciences, 13(3), 230. https://doi.org/10.3390/educsci13030230
(E9) Gabele, M., Weicker, J., Wagner, S., Thoms, A., Huβlein, S., & Hansen, C. (2021). Effects and Ways of Tailored Gamification in Software-Based Training in Cognitive Rehabilitation. https://doi.org/10.1145/3450613.3456828
(E10) Vera, R. R., Arceo, É. E. B., Mendoza, J. C. D., & Pech, J. P. U. (2019). Gamificación para la mejora de procesos en ingeniería de software: Un estudio exploratorio. RECIBE. https://doi.org/10.32870/recibe.v8i1.123
(E11) Deterding, S., Dixon, D. A., Khaled, R., & Nacke, L. E. (2011b). From game design elements to gamefulness. https://doi.org/10.1145/2181037.2181040