A Serious Game with Which to Introduce Users to the World of DevOps

Rubén Grande[a] and Aurora Vizcaíno[b]
Department of Computer Languages and Systems, Universidad de Castilla-La Mancha, Ciudad Real, Spain

Keywords: DevOps, Serious Game, Gamification, Education, Learning.

Abstract: If today's software development organisations are to remain competitive in the software industry, then they need new ways of working or methodologies that will enable them to meet their customers' needs as quickly as possible. Agile methodologies that help organisations to achieve this have been used and developed for several years, but they emphasise only software development teams. However, many teams, such as the IT operations team, are involved in software projects but are not dealt with by agile methodologies, and this is how DevOps came into being. DevOps is a concept that encompasses a new philosophy that seeks to promote the collaboration between the software development team and the IT operations team. However, since it is one of the latest trends in the software industry, there is a lack of training on it, despite the increasing demand for skilled personnel with DevOps knowledge. The goal of this paper is, therefore, to present Journey to the Core of DevOps, a serious game whose objective is to introduce Software Engineering students and inexperienced engineers to the concept of DevOps by providing a virtual environment in which key aspects of DevOps are emphasised, thus allowing players to have a first contact with it.

1 INTRODUCTION

DevOps has, in the last few years, emerged as an important methodology that has been widely adopted in the software industry (Yarlagadda, 2021). Literature provides many definitions and interpretations of DevOps, some of which consider it to be a new methodology for developing software that focuses on automation and changes in the organisational culture (Jabbari, bin Ali, Petersen, & Tanveer, 2016) while others believe it to be the extension of the Agile manifesto to IT operations teams (Lwakatare, Kuvaja, & Oivo, 2016). However, most authors agree that DevOps concerns not only changes in the technologies used, but also changes in culture and methodologies. The benefits that DevOps is able to bring to organisations has been widely studied, and include the shorter delivery time of new software versions, a reduction in costs, less friction between development and operations teams and a better understanding of the software product or service on both sides (Ghantous & Gill, 2017; Jabbari, bin Ali, Petersen, & Tanveer, 2018).

Software organisations are aware of these benefits and are consequently demanding more skilled software engineers who know what DevOps is. They are, therefore, seeking people with both the hard skills required in order to carry out development and operations activities and the knowledge required in order to work in a DevOps environment, thus facilitating its adoption by the business (Hussain, Clear, & MacDonell, 2017). However, higher education organisations are not introducing DevOps into their syllabus, thus leading to a situation in which software companies have difficulty in finding skilled personnel for DevOps. There are several reasons for this lack of education in DevOps: it is difficult to teach it in academic environments, it was not possible to include it in software engineering curricula owing to its innovative nature, time constrains make it unfeasible to teach development and operations skills, and current courses focus solely on theoretical concepts, among others (Bobrov et al., 2020; Pang, Hindle, & Barbosa, 2020).

Educational organisations should, therefore, explore other teaching methods that could be helpful

[a] https://orcid.org/0000-0002-2072-5581
[b] https://orcid.org/0000-0002-0583-6865

Grande, R. and Vizcaíno, A.
A Serious Game with Which to Introduce Users to the World of DevOps.
DOI: 10.5220/0011956400003470
In *Proceedings of the 15th International Conference on Computer Supported Education (CSEDU 2023) - Volume 2*, pages 386-393
ISBN: 978-989-758-641-5; ISSN: 2184-5026

as regards introducing DevOps into their curricula as soon as possible. One of these methods could be the usage of serious games as a tool that would allow students to learn one or more topics while having the freedom to fail (García, Pacheco, León, & Calvo-Manzano, 2020). This method could also benefit software companies as it would allow them to train their employees in DevOps, since they are not usually included in real projects in which failures could mean potential loses to the business.

Despite the promising benefits that serious games could bring to DevOps education, there is, to the best of our knowledge, no serious game with which to learn about DevOps. Our proposal, therefore, consists of a first version of a serious game that helps users to understand some of the concepts identified as being core aspects when learning about DevOps. We are, therefore, of the opinion that this serious game could be useful as regards teaching DevOps to students on software engineering courses and software engineers with no previous knowledge of DevOps.

The remainder of this paper is structured as follows. Previous work is described in Section 2, while the main ideas behind the design of the serious game are presented in Section 3, along with the in-game content. A preliminary validation of the capability of the game to teach the concepts identified is detailed in Section 4, and finally, our conclusions and future work regarding this serious game are addressed in Section 5.

2 RELATED WORKS

This section first examines the serious games related to software engineering and then focuses on the means currently employed to provide education in DevOps.

2.1 Serious Games

The term *serious game* was first defined as a tool whose objective is to educate with an academic purpose while simultaneously entertaining the player of such a game (Abt, 1987). In order to narrow down the scope of serious games to those that are of interest herein, those developed for various software engineering areas will be described. Note that serious games are not mandatory computer-based, such as the described in (Baker, Navarro, & van der Hoek, 2005).

VENTURE (Monasor et al., 2014) is a serious game focused on teaching Global Software Development (GSD). This game enables the player to interact in a virtual development team in which people with different backgrounds and cultures are simulated, thereby enhancing skills such as teamwork or conflict resolution.

GSD-Aware (Vizcaíno, García, Guzmán, & Moraga, 2019) is also focused on GSD, but in this case its objective is to show the player different challenges that an engineer may confront in this type of project, principally as regards communication, coordination and control.

Requenguin (García et al., 2020) was made to support the teaching of the ISO/IEC/IEEE 29148:2011, an international standard that details the processes that needs to be implement by requirements engineering for systems, software products and software services throughout the lifecycle. The game simulates a library where players are asked to apply requirements engineering processes to change the current management system into a software system. The authors concluded, based on the experiment they carried out, that the game not only could potentially improve the acquisition of knowledge of such standard, but also to improve the motivation of the students that played the game.

The objective of Back to Penelope (Marín, Vera, & Giachetti, 2019) is to help to understand effort estimation based on the COSMIC Full Function Point measuring method. In order to achieve this, the player must correctly estimate the development effort required for the vital systems of a spaceship whose astronaut has had an accident and must return to the mothership, which is called Penelope. The game was developed principally in C# by using the game engine Unity. The authors validated that the game allowed 70% of the students to learn how to estimate using the COMISC method.

Riskio (Hart, Margheri, Paci, & Sassone, 2020) is an example of serious game that is not software based. The goal of this tabletop game is to develop cybersecurity awareness in people who work in organisations and do not have a technical background in this subject. The players must, therefore, play the role of both the attacker and the defender of critical assets in a fictitious organisation. The game is based on cards, and the players learn various threats and countermeasures such as spoofing attacks or security training. The authors carried out a study with both employees and students, and analysed the overall perception of the game, perceived ease of use, usefulness and intention to use. The results of this study showed that employees were more confident than were the students that the game could increase their awareness of cybersecurity, but that some changes should be made to the game in order to improve some aspects of it.

2.2 Providing an Education in DevOps

Various teaching tools, challenges and recommendations for the teaching of DevOps can be found in literature. The authors of (Pang et al., 2020) used Grounded Theory to study DevOps education, using different data sources that addressed DevOps education such as academic education, job requirements, certifications, training programmes and DevOps communities, conferences and organisations. In general, the authors concluded that it is difficult to teach DevOps in academic environments owing to the technical knowledge required, along with the difficulty involved in recreating a common environment for development and operations teams. Moreover, the study also highlighted that current courses focus only on the tools used in order to implement DevOps practices.

The work of (Fernandes et al., 2022) analysed the existing field of DevOps education through an interview with 14 DevOps educators. As a result, the authors identified 83 challenges and 185 recommendations, plus relationships among some of them. Around 60% of the challenges identified were specific to the teaching of DevOps itself. Some of these were insufficient time on the course in which to teach DevOps or the large number of DevOps tools. However, some of the recommendations included building scenarios that students could run on their own computers and using alternative learning tools to ease the teaching of DevOps.

A high-level course structure with which to teach DevOps that combines DevOps, Scrum and Challenge Based Learning is proposed in (Cardoso, Chanin, Santos, & Sales, 2022) . The structure proposed has the objective of assisting with the development of both technical and soft skills. Of the soft skills that could be boosted, teamwork, communication and problem solving are highlighted. The authors stated that, in order to assess the effectiveness of the course structure, questionnaires should be applied at the beginning and at the end of the course.

3 SERIOUS GAME DESIGN

This section provides a description of the principles of Journey to the Core of DevOps, which are based on previous games in the software engineering fields, since there is no other serious game for DevOps.

3.1 Core Ideas

It is first necessary to bear in mind that the game audience comprises novice engineers or computer engineering students with poor or no previous knowledge of DevOps. We have, therefore, focused on helping players to learn about some of the topics and practices that were identified as being critical for DevOps.

Some previous literature reviews that have been carried out (Ghantous & Gill, 2017; Jabbari et al., 2016; Leite, Rocha, Kon, Milojicic, & Meirelles, 2019) were used as the basis on which to identify the DevOps concepts that the game should teach: CI/CD pipeline, CI, CD, automation, collaboration and communication.

The principles of the serious game that should guide its development were then defined: a 3D graphics game in which the player will play the role of a developer or an IT operator and will achieve objectives that will make it possible to advance in the game. How these objectives are completed depends on the role chosen, since some roles must actively communicate or collaborate with the other team. As these objectives are achieved, a wall that divides the two teams from each other gradually breaks up, thus making it easier for the teams to visit each other and consequently easing communication and collaboration. Once an objective is achieved, it will affect indicators displayed on the user interface, which is based on the CA(L)MS framework (Kim, Debois, Willis, Humble, & Allspaw, 2016), that will be used to assess the player's performance in the game once it has finished. The acronym of this framework stands for Culture, Automation, Lean, Measurement and Sharing. It is usually used either to assess a company's ability to adopt DevOps processes or as a means to measure success during the adoption of DevOps. There are, therefore, some actions that the player can perform that will increase or decrease the value of one or more of these indicators. In addition, each time players achieve an objective, they are an increase in points is shown on these indicators, which inform them that they are performing correctly.

3.2 Metaphors Used in the Game

In order to support the learning of DevOps, the design of the game has been based on various metaphors, each of which is mapped onto a concept:

- The players first choose the role they wish to play in the game (developer or IT operator) by passing through a door that leads them to the side of the team chosen. These sides are separated by a high

wall representing the traditional separation between development and operations teams (see Figure 3).

- The players then have to interact with various balls that are simplified representations of various work artefacts (usually pieces of code, documentation, databases, statistics, etc.). These interactions consist of creating the artefact, sharing them with the other team or taking them to a given building (see Figure 5).
- CI/CD pipeline. This is considered to be one of the most important technical aspects of DevOps. The metaphor employed to teach this concept consists of a set of buildings that represent the common phases of this pipeline (code repository, integrating the changes, testing, construction, pre-production environment tests and deployment to production). These buildings are identified by means of user-friendly and meaningful icons and tool logos (see Figure 3). If the players choose the developer role, one of their tasks will be to take code balls to the repository, which can be recognised thanks to the GitHub logo. Once this has been done, the pipeline works automatically, and the user can watch the code going through the different phases.
- Automation. An automatic conveyor belt moves the balls through the different phases of the CI/CD pipeline with the objective of allowing the player to understand that no intervention is needed to trigger the tests or to build the software.
- The testing phase is simulated by a scanner that inspects the balls for some seconds. If everything is correct, a green light goes on and the scanner lets the balls advance to the next step.
- The next phase is the build phase, in which all the small balls are transformed into one large ball in order to represent the construction of the software.
- A scanner like that in the testing phase inspects the construction ball in the pre-production environment building, which is on the operations side.
- The construction ball then advances to an elevator that simulates the deployment of the software construction to the production environment, which is represented using various clouds and the AWS logo, indicating that it is a cloud environment (see Figure 7).
- During the game, communication is simulated by promoting interaction with various NPCs (Non playable characters) that are part of either the development team or the operations teams (see Figure 8). These interactions are sometimes necessary in order to achieve objectives. When the player performs an interaction of this nature, a simulated dialogue is represented, and the player is also rewarded with increases in the values on the CALMS indicators.
- Collaboration is simulated when the player is required to visit the opposite side in order to share some balls or discuss work with the NPCs.
- On the IT operations side, there is a customer assistance building that deals with any problems reported by end-users. The IT operations team is, therefore, given this responsibility, and some of the objectives require them to interact with an NPC representing an end-user who has a particular problem.
- The players who play the IT operator role will bring their generated balls to the elevator, in order to represent its direct intervention in the production environment.

The objective of these metaphors is to allow the player to understand the core ideas described previously by representing code with balls that will require specific actions. Moreover, we hope that the icons and logos used may further help the user to understand the meaning of each building in the world of the game. The idea is to abstract the users from technical aspects and focus on the features of DevOps.

Figure 1: Step explaining one of the mechanics of the game.

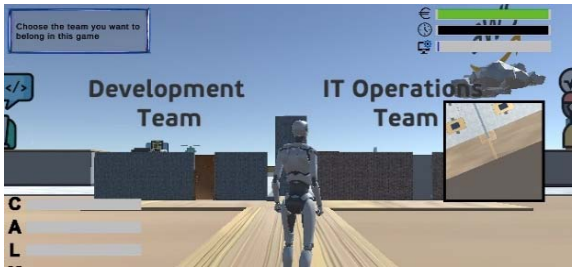

Figure 2: The player must choose a team for the rest of the game.

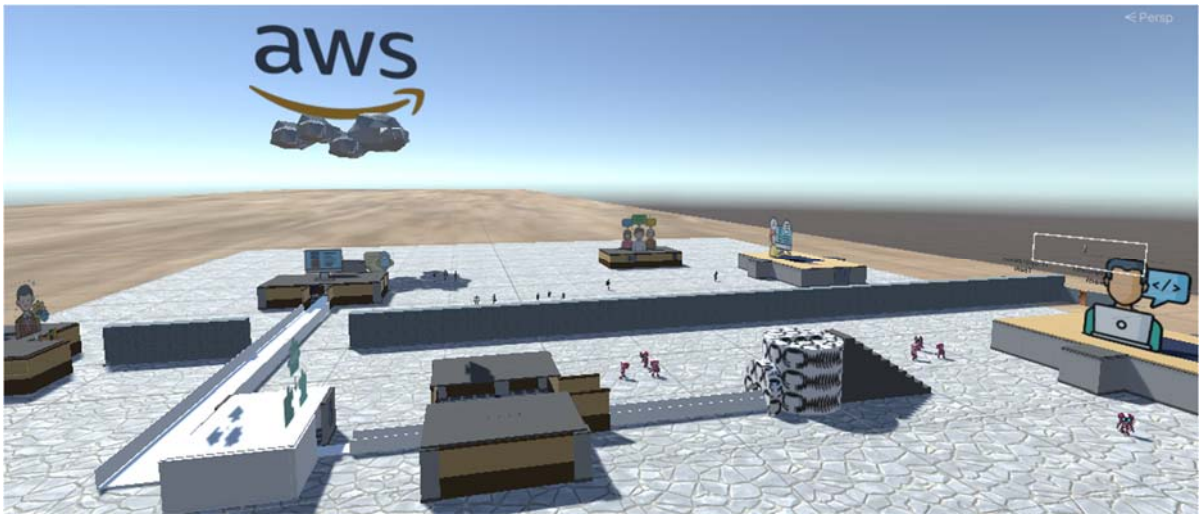

Figure 3: Map of game, in which there are different buildings, metaphors and the CI/CD pipeline.

3.3 Playing the Game

This subsection provides a description of a playthrough of the game. Once the users have chosen the level of difficulty of the game, they will be asked to either watch or skip a tutorial concerning the game. This tutorial instructs the user on what the UI elements mean, the goal of the game and the different game mechanics. Figure 1 shows one of the steps of the tutorial. The world of the game is presented in Figure 3, in which all the relevant buildings can be seen.

The player (the grey avatar in Figure 2) then must choose between being part of the development team or the IT operations team by going through the appropriate gate (see Figure 2).

Once the player has made this decision, the game will start, and each objective will be shown in the upper left-hand corner of the UI. A large red arrow will also appear in order to indicate where the user should go in order to complete the current objective. (see Figure 4).

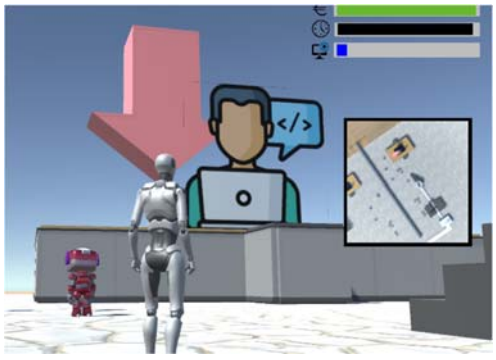

Figure 4: Arrow pointing the development team office.

The player will have to achieve a series of objectives until the blue bar located in the upper right-hand corner (see Figure 4) is filled. This bar gradually fills up every time the player achieves an objective The sequence of objectives requires the user to play the different mechanics of the game: interacting with computers in order to generate balls (see Figure 5), using the CI/CD pipeline (see Figure 6), or communicating with NPCs (see Figure 7), among

Figure 5: Working with a computer to generate a ball.

others. The player should not delay the completion of each objective, since every few seconds the values of the indicators decrease, thus affecting the final evaluation of the player's performance. Once the game ends, statistics are shown regarding how well the player has performed in the game (the final values of the CALMS indicators in percentages), along with tips on how to perform better in subsequent games. This information is saved in order to allow the user to

compare the results obtained in the previous game with those in the latest one.

4 BRIEF EVALUATION

Once the first version of the serious game had been fully developed, we wished to perform a preliminary evaluation of the game with the objective of determining whether it fulfilled the proposed requirements. We also wished to identify the strong and weak points of the version developed in order to

Figure 6: Software build ball being deployed in the elevator to the clouds representing the production environment.

improve it in future versions. These goals were achieved by employing the questionnaire with which to evaluate serious games proposed in (Savi, Wangenheim, & Borgatto, 2011), to which we added some questions related to DevOps concepts. These questions were prepared by following the recommendations of (Kitchenham & Pfleeger, 2002a, 2002b). The questionnaire, therefore, consisted of a form containing 21 questions.

The questionnaire was responded to by 5 students who were at the end of the final year of their software engineering degree. On the one hand, the weak points detected on the basis of the results were:

- The game was not sufficiently immersive.
- There were insufficient game mechanics.

On the other hand, the strong points identified were the following:

- The game helped the users to understand the key aspects of DevOps.
- The game was not monotonous.
- The content of the game was interesting.
- The menus and UI of the game were user-friendly.

Figure 7: Interacting with an NPC on the development team.

This initial evaluation has helped us to focus on improving the degree of immersion of the game, along with recognising the usefulness of the game as regards learning DevOps. However, a more comprehensive and rigorous study is required in order to evaluate whether the metaphors will be understood by a larger number of players. This could also help us to validate the knowledge gain attained by players who know little about DevOps.

5 THREATS TO VALIDITY

The following subsections outline how the different threats to the validity of this study have been dealt with.

5.1 Construct Validity

In order to mitigate this threat, the test employed to evaluate the game was conducted and reviewed by both authors in an attempt to attain different points of view and ask different questions. Moreover, the questions whose objective was to assess the students' learning were as objective as possible, thus

preventing the authors' interpretation from influencing the results.

5.2 Internal Validity

From the perspective of internal validity, two factors were initially considered in the design of this evaluation: the differences between subjects, and participant motivation.

All the participants initially had similar knowledge of DevOps and similar motivation.

5.3 Conclusion Validity

The main threat to the conclusions drawn after validating the serious game described herein could be the relatively low number of students who took the tests. A possible improvement to this evaluation, which is proposed as future work, would be to conduct the study with a larger sample.

5.4 External Validity

It is difficult to generalise the results owing to the small number of participants. However, these findings are preliminary information that will help us to improve the serious game and attain an idea of students' reactions to it.

6 CONCLUSIONS AND FUTURE WORK

The sooner that DevOps is included in the curricula of software engineering and computer engineering courses, the better. However, it is difficult to replicate scenarios in which students have the sense of freedom to fail while they learn the main concepts and practices that form part of DevOps. In addition, changes are made to these study plans on a yearly basis, thus making it more difficult to include technologies and topics that are emerging in industry. There is consequently a need to explore alternative ways in which to address these inconveniences and introduce the learning of DevOps.

This paper describes the first version of a serious game whose objective is to help students to understand the core concepts of DevOps. The mechanics of the game were developed with the goal of not allowing the player's attention to wander, along with it not being difficult to understand. The ideas that the game should help to understand would not, therefore, be overshadowed by these mechanics. The strong point of the game is the metaphors employed, since the mechanics are the drivers that allow the players to discover these metaphors.

After presenting the design of the game, we have described a preliminary evaluation that was carried out with 5 students on the last year of their computer engineering degree. The results obtained initially show that the game helped the participants to understand the concepts described in this paper. This study helped us to identify the strengths of the game, along with the weak points that need to be improved.

Although these results are promising, a more formal study will be carried with a larger number of participants to corroborate that the game could be useful as regards assisting in the learning of DevOps. Participants from different countries will be involved in this study in order to provide us with a sufficiently large number of volunteers, in addition to identifying possible cultural factors that may affect the learning outcomes of the game. This evaluation will be carried with a new version of the game in which the weak points identified in the preliminary evaluation will be addressed.

ACKNOWLEDGEMENTS

This work has been supported by OASSIS project (PID2021-122554OB-C31, funded by MCIN/ AEI / 10.13039/501100011033 / FEDER, EU)

This work is part of the EMMA project "Evaluation and Improvement of the Environmental Sustainability of Applications", (SBPLY/21/180501/000115), funded by the Consejería de Educación, Cultura y Deportes de la Junta de Comunidades de Castilla La Mancha, and FEDER.

REFERENCES

Abt, C. C. (1987). *Serious Games*. University Press of America.

Baker, A., Navarro, E., & van der Hoek, A. (2005). An experimental card game for teaching software engineering processes. *Journal of Systems and Software*, *75*, 3-16. doi: 10.1016/j.jss.2004.02.033

Bobrov, E., Bucchiarone, A., Capozucca, A., Guelfi, N., Mazzara, M., & Masyagin, S. (2020). Teaching DevOps in Academia and Industry: Reflections and Vision. En J.-M. Bruel, M. Mazzara, & B. Meyer (Eds.), *Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment* (pp. 1-14). Cham:

Springer International Publishing. doi: 10.1007/978-3-030-39306-9_1

Cardoso, T., Chanin, R., Santos, A. R., & Sales, A. (2022, diciembre 20). *Combining Agile and DevOps to Improve Students' Tech and Non-tech Skills*. 299-306. Recuperado de https://www.scitepress.org/Link.aspx?doi=10.5220/0010401302990306

Fernandes, M., Ferino, S., Fernandes, A., Kulesza, U., Aranha, E., & Treude, C. (2022). DevOps Education: An Interview Study of Challenges and Recommendations. *2022 IEEE/ACM 44th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, 90-101. doi: 10.1145/3510456.3514152

García, I., Pacheco, C., León, A., & Calvo-Manzano, J. A. (2020). A serious game for teaching the fundamentals of ISO/IEC/IEEE 29148 systems and software engineering – Lifecycle processes – Requirements engineering at undergraduate level. *Computer Standards & Interfaces*, *67*, 103377. doi: 10.1016/j.csi.2019.103377

Ghantous, G. B., & Gill, A. (2017). DevOps: Concepts, Practices, Tools, Benefits and Challenges. *PACIS 2017 Proceedings*. Recuperado de https://aisel.aisnet.org/pacis2017/96

Hart, S., Margheri, A., Paci, F., & Sassone, V. (2020). Riskio: A Serious Game for Cyber Security Awareness and Education. *Computers & Security*, *95*, 101827. doi: 10.1016/j.cose.2020.101827

Hussain, W., Clear, T., & MacDonell, S. (2017). Emerging Trends for Global DevOps: A New Zealand Perspective. *2017 IEEE 12th International Conference on Global Software Engineering (ICGSE)*, 21-30. Buenos Aires, Argentina: IEEE. doi: 10.1109/ICGSE.2017.16

Jabbari, R., bin Ali, N., Petersen, K., & Tanveer, B. (2016). What is DevOps? A Systematic Mapping Study on Definitions and Practices. *Proceedings of the Scientific Workshop Proceedings of XP2016*, 1-11. New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2962695.2962707

Jabbari, R., bin Ali, N., Petersen, K., & Tanveer, B. (2018). Towards a benefits dependency network for DevOps based on a systematic literature review: Towards a benefits dependency network for DevOps based on a systematic literature review. *Journal of Software: Evolution and Process*, *30*(11), e1957. doi: 10.1002/smr.1957

Kim, G., Debois, P., Willis, P. J., Humble, J., & Allspaw, J. (2016). *The DevOPS Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. Portland, OR.

Kitchenham, B. A., & Pfleeger, S. L. (2002a). Principles of survey research part 2: Designing a survey. *ACM SIGSOFT Software Engineering Notes*, *27*(1), 18-20. doi: 10.1145/566493.566495

Kitchenham, B. A., & Pfleeger, S. L. (2002b). Principles of survey research: Part 3: constructing a survey instrument. *ACM SIGSOFT Software Engineering Notes*, *27*(2), 20-24. doi: 10.1145/511152.511155

Leite, L., Rocha, C., Kon, F., Milojicic, D., & Meirelles, P. (2019). A Survey of DevOps Concepts and Challenges. *ACM Computing Surveys*, *52*(6), 127:1-127:35. doi: 10.1145/3359981

Lwakatare, L. E., Kuvaja, P., & Oivo, M. (2016). Relationship of DevOps to Agile, Lean and Continuous Deployment. En P. Abrahamsson, A. Jedlitschka, A. Nguyen Duc, M. Felderer, S. Amasaki, & T. Mikkonen (Eds.), *Product-Focused Software Process Improvement* (pp. 399-415). Cham: Springer International Publishing. doi: 10.1007/978-3-319-49094-6_27

Marín, B., Vera, M., & Giachetti, G. (2019). An adventure serious game for teaching effort estimation in software engineering: 2019 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement, IWSM-Mensura 2019. *CEUR Workshop Proceedings*, *2476*, 71-86.

Monasor, M. J., Parkes, J., Noll, J., Vizcaíno, A., Piattini, M., & Beecham, S. (2014). Global Software Development Education: A Commercial Perspective from a Case Study. *2014 IEEE 9th International Conference on Global Software Engineering*, 173-182. doi: 10.1109/ICGSE.2014.20

Pang, C., Hindle, A., & Barbosa, D. (2020). Understanding devops education with grounded theory. *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering Education and Training*, 107-118. New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3377814.3381711

Savi, R., Wangenheim, C. G. von, & Borgatto, A. F. (2011). A Model for the Evaluation of Educational Games for Teaching Software Engineering. *2011 25th Brazilian Symposium on Software Engineering*, 194-203. doi: 10.1109/SBES.2011.27

Vizcaíno, A., García, F., Guzmán, I. G. R. D., & Moraga, M. Á. (2019). Evaluating GSD-Aware: A Serious Game for Discovering Global Software Development Challenges. *ACM Transactions on Computing Education*, *19*(2), 14:1-14:23. doi: 10.1145/3218279

Yarlagadda, R. T. (2021, marzo 3). *DevOps and Its Practices* [SSRN Scholarly Paper]. Rochester, NY. Recuperado de https://papers.ssrn.com/abstract=3798877.