IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 April 2024, accepted 5 May 2024, date of publication 20 May 2024, date of current version 3 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3403504

APPLIED RESEARCH

Comparing a LEGO® Serious Play Activity With a Traditional Lecture in Software Engineering Education

DANIEL LÓPEZ-FERNÁNDEZ[1], EDMUNDO TOVAR[2], (Senior Member, IEEE), ALDO GORDILLO[1], JOAQUÍN GAYOSO-CABADA[1], CARLOS BADENES-OLMEDO[1], AND ANDREA CIMMINO[1]

[1]ETSI de Sistemas Informáticos, Universidad Politécnica de Madrid, 28031 Madrid, Spain
[2]ETSI Informáticos, Universidad Politécnica de Madrid, 28660 Madrid, Spain

Corresponding author: Edmundo Tovar (edmundo.tovar@upm.es)

This work was supported in part by Universidad Politécnica de Madrid through the Educational Innovation Project under Grant IE23.6102, and in part by the European Union (EU) Program Erasmus+ KA226–Partnerships for Digital Education Readiness under Grant 2020-1-NO01-KA226-VET-094190.

ABSTRACT Active learning methods are needed to promote student motivation and facilitate the development of technical and soft skills. Previous research in software engineering education shows that LEGO® Serious Play (LSP) fully aligns with these needs. However, prior works are usually based on insufficiently robust research methods that do not include a large sample, a variety of evaluation instruments, and/or rigorous comparative methods such as randomized controlled trials, which makes it difficult to obtain reliable and solid conclusions. This article presents an original LSP activity to teach software development life cycle models and core software engineering activities, tackling learning objectives different from those addressed by the LSP activities reported in prior works. The LSP activity was validated through a cluster-randomized controlled trial involving 217 computer science students. These students were divided into a control group that received a traditional lecture and an experimental group that performed the LSP activity. The research was supported by pre and post-tests that allowed the study of the knowledge attained by the students, as well as a questionnaire to gather students' perceptions. The results indicate that the students in the experimental group learned significantly more and were more motivated than their counterparts in the control group. This leads to the conclusion that LSP-based activities such as the one reported in this article are highly effective in terms of knowledge acquisition and motivation to teach some software engineering topics compared to traditional lectures.

INDEX TERMS Software engineering education, LEGO serious play, game-based learning, gamification, active learning, soft skills.

I. INTRODUCTION

Nowadays, it is necessary to combine traditional learning methodologies with active learning approaches that are more appealing to current students. These approaches promote knowledge acquisition as traditional methods do, boost students' motivation, and even enhance soft skills.

A representative active methodology widely employed today in software engineering education with very positive results is game-based learning, which can be applied in different ways, such as through the use of educational or serious card games [1], [2], [3], video games [4], educational escape rooms [5], or virtual reality applications [6]. The LEGO® Serious Play (LSP) methodology is within the game-based learning framework. LSP was designed to improve team performance by facilitating reflection, communication, and

The associate editor coordinating the review of this manuscript and approving it for publication was Nkaepe Olaniyi.

problem-solving [7], and it is grounded on the Serious Play theory [8]. For some years, LSP has been used with very positive results in many knowledge fields, including Software Engineering (SE) [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21].

As will be detailed in the related work section, studies on LSP in software engineering education emphasize the great usefulness of this methodology because it can promote the acquisition of numerous concepts of software engineering, and it can also enhance students' motivation and boost soft skills [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21]. The use of LSP for software engineering education has already been addressed in the last decade. So, what are the contributions of the present work with respect to previous works? Well, let's list them down:

- **Robust methodological design**: Some prior works only describe LSP activities without reporting an empirical validation [10], [12], while most of the works report validations involving small samples and/or scarce research instruments that prevent reliable conclusions from being drawn. We have conducted research involving 227 students, supported by the following evaluation instruments: a pre-test and a post-test to study the acquisition of knowledge and a questionnaire to explore students' perceptions.
- **Rigorous comparison with traditional learning methods**: Most previous work did not compare the effectiveness of LSP versus traditional methods, and those that did [9], [11], [15], [21] incur serious threats to validity arising from small sample size, paucity of evaluation instruments, and/or lack of randomization. We have performed a Randomized Controlled Trial (RCT), the most rigorous research method [22], involving a large sample and several evaluation instruments that allow for a solid comparison of the proposed LSP activity with a traditional lecture.
- **Novel learning objectives**: Except for our previous work [21], most of these contributions deal with learning objectives very different from the one addressed in this article. We have designed an LSP activity to teach the core software engineering activities (i.e., requirements specification, analysis, design, implementation, validation, and maintenance) and several Software Development Life Cycle (SDLC) models. In the LSP activity presented in [21] we addressed some SDLC models (specifically, waterfall, incremental based on evolution, and incremental based on components). The current article also addresses the waterfall model with prototyping, which is very important.

In this article, we address the following **Research Questions** (RQ):

- RQ1: How to perform an LSP activity to teach playfully and actively the waterfall with prototyping software development model?
- RQ2: What are the benefits of knowledge acquisition of using the designed LSP activity in SE education compared to a traditional lecture?
- RQ3: What are the benefits in terms of motivation of using the designed LSP activity in SE education compared to a traditional lecture?

The next sections of the article present the related work, the employed LSP activity, the research methodology, the empirical results and their discussion, and the conclusions.

II. RELATED WORK

Several studies [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21] have been conducted in the last decade to study the usage of LSP in SE education. Overall, these studies emphasize the great usefulness of this methodology since it has the potential to promote the acquisition of numerous software engineering concepts and enhance student motivation and soft skills.

To understand the current state of the art and emphasize the main differentiating elements of the present article, prior studies have been analyzed based on the addressed topic and the following characteristics of the research design: sample size, evaluation instruments, strategies, and comparisons. Table 1 presents the results of this analysis.

Regarding the **SE topics** addressed by prior research, it can be observed that there are a wide variety of topics that can be taught by performing LSP activities: requirements engineering [9], [11], component integration and design of interfaces [10], test-driven development [12], Scrum framework [13], [14], [15], [16], [17], [18], [19], project management [20], or software engineering activities and SDLC models [21].

At this point, it is worthwhile to dwell on one author, S. Kurkovsky, who is especially prolific and has published a lot about activities for learning software engineering through LSP. Beyond his scientific publications [9], [10], [11], [12], he has also posted on the web a series of LSP activities with its corresponding instructions ready to be used by other teachers or trainers [23]. As regards non-scientific publications, it is also worth mentioning the book Lego4Scrum by A. Krivitsky [24], which presents a very popular LSP activity named Lego City, the use of which is reported in several scientific publications [13], [14], [15], [16].

Regarding the empirical validation, some previous works only describe LSP activities without reporting an empirical validation [10], [12], but most of them report an LSP activity and a validation. However, most of these works [9], [11], [13], [14], [15], [16], [17], [19], [20] involve such a small **sample size** that reliable and generalizable conclusions cannot be drawn. Only two articles report studies involving a large sample size (more than 60 participants) [18], [21], but they suffer from some problems mentioned below.

Leaving aside the sample size problem, many of the aforementioned works incur other methodological problems. The most critical is related to **evaluation instruments**. Some authors do not use any evaluation instrument beyond direct observation [14], others only use interviews [16], others only use a perceptions questionnaire [9], [13], [18], [20], and others combine interviews and questionnaires [15]. The evaluation relies exclusively on subjective self-reported student data in all these cases. Only three works use more sophisticated evaluation instruments that provide more objective results, such as validated tests to measure motivation [11] or theoretical tests to measure knowledge acquisition [17], [19].

The limitations presented by some of these works are quite illuminating with respect to the usual problems regarding evaluation instruments found in prior research about LSP on SE education. For example, in [13] it is stated that "*We cannot make any precise claims about learning because we have not taken into account any assessment in this report"* or *"There are also potential flaws in the instrument used to collect data*[a student's perception questionnaire] *since it was not initially intended to gather a precise evaluation of learning, but rather a notion of their engagement…*". This shows that in many cases the authors of papers such as these focus on reporting an LSP activity, which may be innovative and interesting, rather than rigorously demonstrating the real effectiveness of the activity.

Moreover, some works [9], [11], [15], [21] compare **LSP activities with traditional learning methods**, and some of them [9], [11] even do so by randomly assigning students to the control and experimental groups. However, the aforementioned problems related to sample size and/or the scarce use of evaluation instruments greatly limit the validity of the conclusions derived from this comparison. In addition, some studies compare LSP activities with other learning methods such as educational virtual reality games [19] or video-based learning [20], but those studies also suffer from the above-mentioned problems.

The only prior work that has a large sample size and uses evaluation instruments beyond perceptions questionnaires, and compares the effectiveness of an LSP activity with a traditional method is the one presented by some of the authors of the present article [21]. However, that article has several limitations that threaten the validity of the conclusions obtained: the assignment of students to control and experimental groups was not random and the distribution was very unbalanced (58 vs. 184), the students did not complete a pre-test, so the knowledge acquisition achieved by the involved students could not be measured accurately, the post-test only contained five questions and certain theoretical concepts were not asked about, and the students of the control group were not surveyed.

The current article proposes a new LSP activity and aims to overcome the methodological limitations raised in the presented analysis to obtain reliable, solid, and generalizable conclusions.

III. PRESENTATION OF THE LEGO® SERIOUS PLAY ACTIVITY

This activity aims to teach the core software engineering activities and some SDLC models. Moreover, the activity is also intended to make students think about specific common problems in software engineering, such as vague requirements, deficient traceability, or maintenance management. Lastly, the activity aims to enhance soft skills such as leadership, teamwork, and communication. In this regard, the activity responds to the skills needs priorities identified in the STEM graduates group carried out under the STEMSOFT Project, taking the "Skillsbank" framework as a reference, which is a STEM competence map-oriented [26].

The LSP activity is as follows. The teacher briefly explains the core activities of software engineering and some representative SDLC models. Next, the teacher introduces the activity, and the students form teams of about five people. The teacher did not define the roles of the students in the team to achieve a more organic teamwork. In this way, the students organized themselves, internally assigning appropriate roles. During the activity, the teacher observed and promoted the team's productive functioning and said that all team members were involved in completing the tasks.

The teams of students should build a chair made with LEGO® pieces by going through the core activities of software engineering in a certain way that determines the SDLC model they are using. In this way, the activities performed using the LEGO® materials are fully integrated with learning the SE activities and the SDLC models. During the execution of the activity, the teacher controls the process and sometimes acts as a client. The prior version of the LSP activity addresses the following SDLC models: waterfall, incremental based on evolution, and incremental based on components. Some examples of the chairs that were built with LEGO® bricks are depicted in Figure 1. More details about the activity can be found in [21].

FIGURE 1. LEGO® chairs built during the LSP activity.

The new version of the LSP activity keeps the essence of the prior version and incorporates a new SDLC model:

TABLE 1. Analisys of the state of the art: LSP on software engineering education.

Study	SE Topic	Empirical validation	Sample size	Evaluation instruments	Pre-post strategy (instrument)	Comparison with other methods (method)	Comparison based on randomization
[9]	Requirements engineering, systems dependability	YES	20	Perceptions questionnaire	NO	YES (lecture)	YES
[10]	Component integration and design of interfaces	NO	-	-	-	-	-
[11]	Requirements engineering	YES	21	Motivated Strategies for Learning Questionnaire (MSLQ) [25], course exam (two questions), interview	YES (MSLQ)	YES (lecture)	YES
[12]	Test-driven development	NO	-	-	-	-	-
[13]	Requirements engineering, Scrum framework	YES	27	Perceptions questionnaire	NO	NO	-
[14]	Scrum framework	YES	24	Observation	NO	NO	-
[15]	Scrum framework	YES	46	Perceptions questionnaire, learning diary, interview	NO	YES (lecture)	NO
[16]	Scrum framework	YES	44	Interview	NO	NO	-
[17]	Scrum framework and design thinking	YES	24	Pre-test, post-test, and motivation test	YES	NO	-
[18]	Scrum framework	YES	198	Perceptions questionnaire	NO	NO	-
[19]	Scrum framework	YES	59	Pre-test, post-test, perceptions questionnaire	YES	YES (Virtual reality)	NO
[20]	Project management	YES	44	Perceptions questionnaire	NO	YES (Video-based learning)	NO
[21]	Software engineering activities and SDLC models	YES	242	Post-test (five questions), perceptions questionnaire [only in experimental group]	NO	YES (lecture)	NO

the waterfall model with prototyping. In this model, the students should build a chair by carrying out the software engineering core activities sequentially and supplementing the requirements specification with a prototype made with plasticine. The activities performed in this model are addressed as follows:

1) Requirements elicitation and prototyping: Students first describe the chair's requirements in writing. Then, they make a chair prototype with plasticine, which the teacher, acting as the client, validates. Lastly, students can modify the initial requirements based on the client's feedback.
2) Analysis: Students represent the chair in an isometric perspective. Since the analysis does not go into the implementation details, the drawing should not consider the exact number of blocks.
3) Design: Students represent the chair's floor, elevation, and profile plans. Since the design details implementation, the drawings should represent the exact number of blocks.
4) Implementation: Students select the LEGO® pieces specified in their design drawings and build the chair following them.
5) Validation: The teacher, again acting as a client, checks that the product is suitable for a LEGO® figure and provides feedback. Moreover, he/she checks the traceability of the outcomes produced so far.
6) Maintenance and evolution: The teacher may "fortuitously" break the chair while revising it, and students must repair it (maintenance). Also, it may be necessary if the teacher demands new functionalities or features for the chair evolution.

Finally, the students achieved a chair built with LEGO® bricks and a plasticine prototype that should have helped them understand the client's needs quickly and build a better chair. Figure 2 depicts an example of a plasticine prototype, a LEGO® chair validated by a user (i.e., a LEGO® figure), and the "diagrams" made in the analysis and design phases.

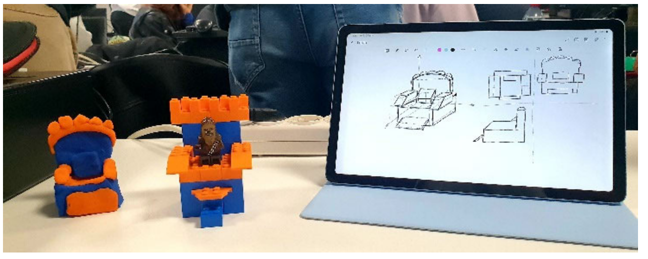


FIGURE 2. Outcomes of the LSP activity when using waterfall with the prototype as an SDLC model.

IV. RESEARCH METHODOLOGY

This section describes the cluster-randomized controlled trial used to study and compare the learning outcomes of the LSP activity with a traditional lecture. It should be noted that this research methodology is widely used in education research, and it is a type of randomized controlled trial in which the unit of randomization is not the participant but a previously defined group [27]. In this case, the unit of randomization was the class groups previously organized in the course, and the whole learning experience was carried out within a period of no more than 24 hours. Some class groups performed the LSP activity (forming the experimental group), whereas others received a traditional lecture (forming the control group).

A. CONTEXT AND SAMPLE

This research was conducted during the academic year 2022-23 at the Faculty of Computer Science from the Universidad Politécnica de Madrid. Specifically, the research was conducted in a fourth-semester Software Engineering Fundamentals course. This course covers, among other topics, SDLC models and the core software engineering activities. This research was conducted on the introductory topic, which covers SDLC models and a brief introduction to the core software engineering activities.

This course had so many students enrolled that the faculty divided them into different class groups, which were used as clusters in the research. These clusters resulted in a control group of 110 students and an experimental group of 117 students. The control group students attended a traditional lecture about SDLC models and the core software engineering activities, while the experimental group students performed the LSP activity. The students were randomly divided into the experimental group to ensure that all SDLC models were addressed equally.

B. PROCEDURE

First, all participating students gave informed consent to participate in this research. Then, all the students were given 10 minutes to complete the pre-test. After that, students in the control group attended the traditional lecture for approximately one hour, while students in the experimental group performed the LSP activity, which lasted around one hour. Then, all students were given 10 minutes to take the post-test. Lastly, the students completed the perceptions questionnaire voluntarily and anonymously about the learning methodology they experimented with. The entire intervention lasted around one hour and a half in both groups, including completing tests and the questionnaire. All participating students completed both the pre-and post-test, and most of these students also completed the perceptions questionnaire (94 in the control group and 114 in the experimental group).

C. METHODS AND INSTRUMENTS

A pre-test and a post-test were utilized to measure the knowledge acquisition achieved by the students in both groups. These tests were scored from 0 to 10 and had the same ten theoretical questions about the targeted topics. After completing the pre-test, the students did not receive feedback to prevent them from memorizing the answers, which could jeopardize the post-test results. Moreover, the test scores did not affect the final grades of the students to discourage unexpected behaviors.

Furthermore, a questionnaire was also employed to collect student opinions about the learning methodologies under study. The questionnaire used in both groups contained the same seven questions, but the questionnaire in the experimental group contained two additional questions not applicable to the control group. The questions were evaluated using a Likert scale from 1 (total disagree) to 5 (total agree). The questions are shown together with the results.

D. DATA ANALYSIS

Firstly, it was found through a Shapiro-Wilk test that the data collected from the tests followed a normal distribution and that it was suitable to use parametric tests. Student's T test for paired samples was employed to find statistically significant differences between the scores achieved by the students in both tests. Furthermore, the Student's T test for independent samples was employed to find statistically significant differences between the scores achieved in the test by the students of the control group and those achieved by the students of the experimental group. Moreover, Cohen's d was used as the effect size measure in the comparisons, and Cohen's guidelines were used to interpret the resulting values as small effect size ($d = 0.2$), medium effect size ($d = 0.5$), or large effect size ($d \geq 0.8$).

Second, it was found through a Shapiro-Wilk test that the data collected from the questionnaire followed a non-normal distribution and that, in this case, it was suitable to use non-parametric tests. The Mann-Whitney test for independent samples was used to compare students' perceptions of both groups. Moreover, the r correlation coefficient was used to measure the comparison effect size. Cohen's guidelines were also used to interpret the resulting values as small effect size ($r = 0.1$), medium effect size ($r = 0.3$), or large effect size ($r \geq 0.5$).

Finally, descriptive statistics (mean and standard deviation) were employed to analyze the test and questionnaire results.

V. RESULTS

A. LEARNING PERFORMANCE

Table 2 depicts the results obtained from the tests realized by the students, the resulting learning gains (calculated as the difference between post-test and pre-test scores), and all the comparisons carried out.

The control and experimental groups achieved learning gains of 1.65 and 2.33, respectively. This improvement is statistically significant in both cases with a large effect size (see column "Paired samples T-test"). These results reveal that both learning approaches were useful for knowledge acquisition.

Furthermore, a statistically significant difference was not found in the pre-test scores, thus confirming that initial knowledge about the covered topic was similar in both groups. This fact confirms that the clusters used to configure the experimental and control groups were homogeneous.

In addition, statistically significant differences with a medium effect size were identified between both groups in post-test scores and learning gains (see row "Independent samples T-test"). This indicates that the LSP activity in the experimental group outperformed the traditional lecture carried out in the control group regarding knowledge acquisition.

TABLE 2. Learning performance results.

		Pre-test	Post-test	Learning gains	Paired samples T-test	
		M (SD)	M (SD)	M (SD)	p-value	Effect Size (d)
Control Group (N=110)		5.30 (1.46)	6.96 (1.40)	1.65 (2.04)	< 0.001	0.82
Experimental Group (N=117)		5.04 (1.52)	7.37 (1.31)	2.33 (1.68)	< 0.001	1.16
Independent samples T-test	p-value	0.18	0.02	0.01	-	
	Effect Size (d)	0.17	0.30	0.36		

B. STUDENT'S PERCEPTIONS

Table 3 depicts the items of the questionnaire and the scores provided by each group of students after the experience. As can be seen, the scores obtained in both groups are positive (above 4.3 out of 5), especially in the experimental group. Indeed, among the 7 items compared, statistically significant differences ($p < 0.05$) favorable towards the experimental group were identified in 5 of them (see column "Mann-Whitney Test for Independent Samples").

The students considered both methodologies helpful in learning the concepts addressed in the experience. However, in the other aspects evaluated, the students rated the LSP activity significantly more positively: general opinion, enhancement of the learning process, fun, motivation, and acceptance.

Moreover, the experimental group students considered that the LSP activity was also effective in developing soft skills, a dimension uncovered by the traditional lecture. Finally, the students in the experimental group found the LSP activity a very good complement to traditional methodologies.

Overall, the results obtained from the questionnaire show that the LSP activity is more beneficial than the traditional lecture, both from the student's point of view and from the instructor's point of view.

TABLE 3. Questionnaire results.

Item	Control Group (N=92)	Experimental Group (N=114)	Mann-Whitney Test for Independent Samples	
	M (SD)	M (SD)	p-value	Effect Size (r)
My opinion about the activity is positive.	4.55 (0.59)	4.88 (0.35)	< 0.001	0.32
The activity enhanced my learning process	4.50 (0.58)	4.70 (0.50)	0.008	0.18
The activity was useful to learn the core activities in software engineering	4.56 (0.60)	4.68 (0.54)	0.16	0.10
The activity was useful to learn software engineering life cycle models	4.48 (0.65)	4.59 (0.60)	0.21	0.08
The activity was fun and entertaining	4.42 (0.76)	4.85 (0.43)	< 0.001	0.33
The activity was motivating and appealing	4.42 (0.71)	4.77 (0.48)	< 0.001	0.28
I would like to learn with activities like this in the future	4.37 (0.80)	4.88 (0.38)	< 0.001	0.38
The LSP activity was helpful to develop soft skills	-	4.61 (0.66)	-	-
The LSP activity is a good complement to traditional methodologies	-	4.75 (0.48)	-	-

VI. DISCUSSION

This section contains a discussion of the RQs addressed in the research. Regarding RQ1 (*How to perform an LSP activity to teach playfully and actively the waterfall with prototyping software development model?*), a detailed presentation of the proposed LSP activity to teach playfully and actively the waterfall with prototyping software development model is presented in Section III. Moreover, the results indicate that the students in the experimental group (who performed the LSP activity) learned significantly more than their counterparts in the control group (who received a traditional lecture) and found this learning more motivating. This allows us to answer RQ2 (*What are the benefits in terms of knowledge acquisition of using the designed LSP activity in SE education compared to a traditional lecture?*) and RQ3 (*What are the benefits in terms of motivation of using the designed LSP activity in SE education compared to a traditional lecture?*) as detailed below.

The results show that the teaching methods (i.e., traditional lecture and LSP) are effective in software engineering education regarding knowledge acquisition and motivation, at least in teaching core software engineering activities and SDLC models. The effectiveness of the knowledge acquisition perspective can be appreciated in the results regarding the learning performance, which reveal statistically significant differences with a large size effect between the knowledge tests taken by the students before and after the traditional lecture (learning gains = 1.65, p-value < 0.001, d = 0.82) and the LSP activity (learning gains = 2.33, p-value < 0.001, d = 1.16). The effectiveness of the motivational perspective can be appreciated in the results regarding the student's perceptions, which indicated a very positive appreciation of both learning methods (every item was rated above 4.3 over 5).

This finding is not very novel since the effectiveness of traditional methods has been demonstrated for a long time. Moreover, in recent years, many empirical studies have indicated that LSP activities are helpful in software engineering education [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20]. These studies did not address the same learning objectives as the LSP activity presented in this contribution but showed the efficacy of LSP activities in addressing essential subjects such as requirements engineering [9], [11], [13], the Scrum framework [13], [14], [15], [16], [17], [18], [19], or project management [20]. However, these contributions based their findings on studies with a much less robust design and/or a much smaller sample than the study presented in the current contribution. The current study consists of a cluster-randomized controlled trial involving pre-and post-tests, perception questionnaires, and a sample of 227 students. Therefore, this contribution represents a further step in this area of research.

Beyond this, the results obtained in this research also show that the LSP methodology outperformed the traditional lecture in terms of the acquisition of knowledge and other important aspects, such as motivation, fun, and the improvement of soft skills, which are of great importance in engineering education [28]. The improvement of the knowledge acquisition dimension (related to RQ2) can be appreciated in the results regarding the learning performance, which reveal a statistically significant difference with a small-to-medium effect size between the learning gains attained by the students who received the traditional lecture and those who performed the LSP activity (1.65 vs 2.33, p-value = 0.01, d = 0.36). The improvement of the motivational dimension (related to RQ3) can be appreciated in the results regarding the student's perceptions, which reveal five items (out of 7) with significant statistical differences ($p < 0.05$) and small or medium size effect ($r > 0.10$) favorable to the LSP activity. These items evaluate aspects such as general opinion (4.55 vs. 4.88, p-value < 0.001, r= 0.32), enhancement of the learning process (4.50 vs. 4.70, p-value = 0.008, r= 0.18), fun (4.42 vs. 4.85, p-value < 0.001, r= 0.33), motivation (4.42 vs. 4.77, p-value < 0.001, r= 0.28), and acceptance (4.37 vs 4.88, p-value < 0.001, r= 0.38).

Our previous research [21] showed that the students who conducted an LSP activity similar to the one presented in the current article seemed to learn a little more than students who received a traditional lecture, but statistically significant differences were not found. However, the knowledge evaluation instrument used in that research contained few questions, and since no pre-post strategy was applied, no reliable measurement of learning gains could be obtained. Moreover, the motivational dimension could not be compared since the student's perceptions questionnaire was not used in the control group.

Therefore, the current contribution has overcome these limitations and presents a sufficiently robust study that affirms that teaching based on LSP can be more effective in terms of knowledge acquisition and motivation than traditional teaching in software engineering education, at least in the teaching of the core software engineering activities and SDLC models. This is the premier finding of this contribution because, to the authors' best knowledge, this is the first work that proves in such a convincing way the effectiveness of LSP in software engineering education.

This finding is because, in general, learning through the LSP methodology is more active and experiential than learning through a traditional methodology. In addition, the topic addressed by this particular research is explained in a very illustrative and practical way through the presented LSP activity because students can experience first-hand the core activities of software engineering and become aware of the key features of each SDLC model through very illustrative metaphors.

VII. CONCLUSION

This article presents an original LSP activity to learn the core software engineering activities and some SDLC models. The results prove this LSP activity is more effective than a traditional lecture. Although the traditional lecture is effective, the LSP activity has achieved significantly better results. The students who carried out the presented LSP activity achieved more significant learning gains and greater motivation than those who attended the traditional lecture. In addition, those students could train their soft skills through the LSP activity, while those who participated in the traditional class did not.

This finding, together with those obtained from previous related work, leads to the conclusion that LSP-based activities are highly effective in knowledge acquisition and motivation to teach some software engineering topics, even more so than classic learning activities like traditional lectures. They also enable training soft skills that are difficult to address through traditional methods.

Nevertheless, despite the robustness of the present study, it is not without limitations, and there are specific threats to validity that may limit the generalizability of the reported conclusions. First, we only analyzed one LSP activity, and the obtained results are not necessarily generalizable to all LSP activities because each has a series of particularities and, hence, a different learning effectiveness. Second, we conducted the study in a very concrete part of the world (specifically in Spain, Europe). The obtained results are not necessarily generalizable to SE students from cultures very different from those of the students in the study sample since LSP activities involve soft skills such as teamwork and communication, which are culture-dependent.

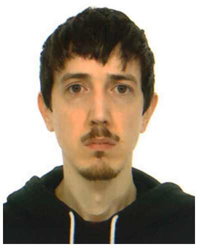
Therefore, we deem that more robust research on existing and new LSP activities is needed to address these limitations. Moreover, we believe that studies like those presented in this paper should be replicated in different parts of the world. Future work could also address comparisons of the designed LSP activity with other innovative learning alternatives and the design of new LSP activities to teach other topics of interest in SE education, such as software architecture or DevOps.

REFERENCES

[1] A. Baker, E. Oh Navarro, and A. van der Hoek, "An experimental card game for teaching software engineering processes," *J. Syst. Softw.*, vol. 75, nos. 1–2, pp. 3–16, Feb. 2005.
[2] D. Carrington, A. Baker, and A. van der Hoek, "It's all in the game: Teaching software process concepts," in *Proc. Frontiers Educ. 35th Annu. Conf.*, Oct. 2005, p. F4G.
[3] J. M. Fernandes and S. M. Sousa, "Playscrum—A card game to learn the scrum agile method," in *Proc. 2nd Int. Conf. Games Virtual Worlds Serious Appl.*, Mar. 2010, pp. 52–59.
[4] D. López-Fernández, A. Gordillo, P. P. Alarcón, and E. Tovar, "Comparing traditional teaching and game-based learning using teacher-authored games on computer science education," *IEEE Trans. Educ.*, vol. 64, no. 4, pp. 367–373, Nov. 2021.
[5] A. Gordillo, D. López-Fernández, S. López-Pernas, and J. Quemada, "Evaluating an educational escape room conducted remotely for teaching software engineering," *IEEE Access*, vol. 8, pp. 225032–225051, 2020.
[6] D. López-Fernández, J. Mayor, J. Pérez, and A. Gordillo, "Learning and motivational impact of using a virtual reality serious video game to learn scrum," *IEEE Trans. Games*, vol. 15, no. 3, pp. 430–439, Sep. 2023.
[7] LEGO. (2023). *LEGO Serious Play*. [Online]. Available: https://www.lego.com/en-us/seriousplay (URL)
[8] J. Roos and B. Victor, "How it all began: The origins of LEGO serious play," *Int. J. Manag. Appl. Res.*, vol. 5, no. 4, pp. 326–343, 2018.
[9] S. Kurkovsky, "Teaching software engineering with LEGO serious play," in *Proc. ACM Conf. Innov. Technol. Comput. Sci. Educ.*, 2015, pp. 213–218.
[10] S. Kurkovsky, "Using LEGO to teach software interfaces and integration," in *Proc. 23rd Annu. ACM Conf. Innov. Technol. Comput. Sci. Educ.*, 2018, pp. 371–372.
[11] S. Kurkovsky, S. Ludi, and L. Clark, "Active learning with LEGO for software requirements," in *Proc. 50th ACM Tech. Symp. Comput. Sci. Educ.*, 2019, pp. 218–224.
[12] S. Kurkovsky, "A LEGO-based approach to introducing test-driven development," in *Proc. ACM Conf. Innov. Technol. Comput. Sci. Educ.*, vol. 2016, pp. 246–247.
[13] K. Gama, "An experience report on using LEGO-based activities in a software engineering course," in *Proc. 33rd Brazilian Symp. Softw. Eng.*, 2019, pp. 289–298.
[14] M. Velić, I. Padavić, and Ž. Dobrović, "Metamodel of agile project management and the process of building with LEGO bricks," in *Proc. 23rd Central Eur. Conf. Inf. Intell. Syst. (CECIIS)*, 2012, pp. 193–481.
[15] M. Paasivaara, V. Heikkilä, C. Lassenius, and T. Toivola, "Teaching students scrum using LEGO blocks," in *Proc. Companion 36th Int. Conf. Softw. Eng.*, 2014, pp. 382–391.
[16] M. Kropp, A. Meier, M. Mateescu, and C. Zahn, "Teaching and learning agile collaboration," in *Proc. IEEE 27th Conf. Softw. Eng. Educ. Training (CSEE&T)*, Apr. 2014, pp. 139–148.
[17] C. Villarrubia, J. M. Vara, D. Granada, C. Gómez-Macías, and F. J. Pérez-Blanco, "DesignScrum—An agility educational resource powered by creativity," *Softw., Pract. Exper.*, vol. 54, no. 6, pp. 985–1009, Jun. 2024.
[18] S. Bourdeau, A. Romero-Torres, and M. C. Petit, *Learning Scrum: A LEGO®-Scrum Simulation in Agile Scrum Implementation and Its Long-Term Impact on Organizations*. PA, USA: IGI Global, 2021, pp. 169–189.
[19] A. Gordillo, D. López-Fernández, and J. Mayor, "Examining and comparing the effectiveness of virtual reality serious games and LEGO serious play for learning scrum," *Appl. Sci.*, vol. 14, no. 2, p. 830, Jan. 2024.
[20] A. Alzaghoul, E. Tovar, A. Rodriguez-Sevillano, and M. Barcala-Montejano, "Comparison between video-class and LEGO serious slay learning strategies for the students of engineering discipline," *Int. J. Eng. Educ.*, 36, no. 1, pp. 256–266, 2020.
[21] D. López-Fernández, A. Gordillo, F. Ortega, A. Yagüe, and E. Tovar, "LEGO serious play in software engineering education," *IEEE Access*, vol. 9, pp. 103120–103131, 2021.
[22] L. M. Woolfson, *Educational Psychology, the Impact of Psychological Research on Education*. Upper Saddle River, NJ, USA: Prentice-Hall, 2011.
[23] S. Kurkovsky. (2024). *LEGO for Software Engineering: Active Learning Exercises for Classroom Use*. [Online]. Available: https://web.ccsu.edu/lego-se/
[24] A. Krivitsky, "Lego4scrum: A complete guide to LEGO4scrum—A great way to teach the Scrum framework and Agile thinking," in *Amazon Digital Services LLC*, 3rd ed. Independently Published, 2017.
[25] P. R. Pintrich and E. V. de Groot, "Motivational and self-regulated learning components of classroom academic performance," *J. Educ. Psychol.*, vol. 82, no. 1, pp. 33–40, 1990.
[26] A. Friesel, E. Tovar, L. Previati, E. Vizzarro, L. Erik Erikssøn, and P. Binelli, "Empowering STEM candidates for employability and entrepreneurship," in *Proc. 32nd Annu. Conf. Eur. Assoc. Educ. Electr. Inf. Eng. (EAEEIE)*, Eindhoven, The Netherlands, Jun. 2023, pp. 1–6, doi: 10.23919/EAEEIE55804.2023.10181377.
[27] S. Puffer, D. J. Torgerson, and J. Watson, "Cluster randomized controlled trials," *J. Eval. Clin. Pract.*, vol. 11, no. 5, pp. 479–483, 2005.
[28] D. L. Fernández, V. Lapuerta, and M. Casado, "Socio-emotional competences at university: Optimization of learning and professional competitiveness of engineering students," *Int. J. Eng. Educ.*, vol. 31, pp. 33–41, 2015.

DANIEL LÓPEZ-FERNÁNDEZ received the degree in software engineering and the Ph.D. degree in software and systems from Universidad Politécnica de Madrid (UPM). He is currently an Associate Professor with the Computer Science Department, UPM. He also leads the Educational Innovation Group of Active Learning Methods and Educational Technologies (GIETEMA). His research interests include the application of active learning methods, the development of innovative educational tools, and the study of agile methodologies in software companies. He received the Educational Innovation Award from UPM.

EDMUNDO TOVAR (Senior Member, IEEE) received the degree in computer engineering and the Ph.D. degree in informatics from Universidad Politécnica de Madrid (UPM). He is currently a Professor with the Computer Languages and Systems and Software Engineering Department, UPM. He is a professional member of IEEE ETA KAPPA NU. He was a member of the IEEE Education Society Board of Governors (2005–2012), where he was the President-Elect (2019–2020) and the President (2021-2022). He leads the Innovation Group in Technologies Applied to Open Education. He is an Editor of *Information Technology Journal*/*Information Systems Journal* and an Editorial Board Member of MERLOT.



ALDO GORDILLO received the degree in telecommunications engineering and the Ph.D. degree in telematics engineering from Universidad Politécnica de Madrid (UPM). He is currently an Associate Professor with the Computer Science Department, UPM. His research interests include the field of technology-enhanced learning, with a special focus on the creation, evaluation, and dissemination of e-learning resources, computing education, game-based learning, gamification, and e-learning systems.

JOAQUÍN GAYOSO-CABADA received the degree in computer science and the Ph.D. degree in software and systems from Universidad Complutense de Madrid. He is currently an Associate Professor with the Computer Science Department, UPM. His research interests include data transformation and retrieving, extraction of underlying information, digital repositories, digital humanities, and game-based learning.

CARLOS BADENES-OLMEDO received the degree in computer science and the Ph.D. degree in artificial intelligence from Universidad Politécnica de Madrid (UPM). He is currently an Associate Professor with the Computer Science Department, UPM. His research interests include AI-based algorithms for large-scale document management and software engineering education.

ANDREA CIMMINO received the degree in computer science and the Ph.D. degree in software engineering from Universidad de Sevilla (US). He is currently an Associate Professor with the Computer Science Department, Universidad Politécnica de Madrid (UPM). His research interests include the semantic web, semantic interoperability, data integration, and the IoT discovery.