Drawing Based Game for Teaching Empirysm and Continuous Improvement in Scrum

Krzysztof Marek
Warsaw University of Technology - Faculty of Electrical Engineering
Warsaw, Poland *krzysztof.marek@pw.edu.pl*

Kamila Martyniuk-Sienkiewicz
Warsaw University of Technology - Faculty of Civil Engineering
Warsaw, Poland *kamila.sienkiewicz@pw.edu.pl*

Abstract

Empiricism and continuous improvement remain the backbone of Scrum and other Agile methods. However, building a practical understanding of such a process during a one-day Scrum workshop or a university course remains an ongoing challenge. This paper introduces an extended version of a novel drawing-based game designed to demonstrate the use of client feedback in continuous improvement process and simulating work within a Scrum framework by means of an analogy involving the drawing of icons as an alternative to randomly generated results. In the performed study, 107 participants in subgroups took part in a one-hour-long game session and solved a Scrum knowledge test before and after each game. Results show noticeable improvement in Scrum knowledge among participants. Additionally, the drawing aspect of the game has been found engaging and significantly enhanced the construction of analogies between gameplay and real-world software development processes during the game and further work with the participants. This research adds to the existing knowledge on Scrum coaching and teaching, providing a simple-to-set-up game allowing for the simulation of the use of empiricism and continuous improvement without the need for complex, expensive tools or environments.

Keywords: Scrum, Gamification, Scrum Coaching, Educational games

1. Introduction

Scrum remains a predominant agile approach [5] in software development, making the need to train new team members in the framework both at the university level and in companies, even more important. Traditional teaching in the form of direct explanation of the given topic and providing step-by-step guidelines struggles in today's world, with approaches blending coaching into the traditional teaching becoming more popular [14]. In case of Scrum numerous training initiatives remain focused predominantly on the transmission of theoretical knowledge, often emphasizing terminology without fostering a deeper comprehension or practical engagement with key Scrum elements such as emphirism and continuous improvement, Unfortunately the introduction of natural coaching into a university course or a one- or two-day training workshop is challenging due to the complexity of the software development process, leaving the newcomers less prepared for future work in a Scrum team.

Given the difficulties associated with conveying the practical aspects of Scrum, many training workshops have incorporated elements of gamification into their educational strategies. Gamification refers to the use of game-derived elements (such as point scoring, challenges, and progress tracking) in non-game contexts to enhance participant engagement and motivation. Among the widely recognized Scrum-based games are LEGO4Scrum [11], which utilizes LEGO bricks, and the card-based PlayScrum [6]. Considering that Scrum practitioners commonly integrate several gamified elements (e.g., Planning Poker), gamification is a particularly effective approach for teaching Scrum. Nonetheless, many of these games require access to specific materials (e.g., LEGO sets, customized cards) and are often designed to concentrate on selected aspects of Scrum or focus on remembering the names and steps in the framework rather than experiencing the empirical process (transparency-> inspection ->adaptation) as a whole.

In response to these observations, the primary objective of this paper is to present a game simulating the entire Scrum empirical software engineering process derived from an existing teaching game using drawing to teach Scrum with the focus on quality and Definition of Done [15]. The new extended version's goal is to convey the Scrum-based empirical software development process in a straightforward manner without the need for specialized equipment. The game engages participants, organized into small teams, in executing Sprints to complete designated Product Backlog Items (PBI). As in the original game, these PBI involve drawing icons based on provided requirements. However, the proposed new version additionally takes into account the business scenario, introducing the element of prioritisation and optimization of business value delivery, allowing participants to perform a full Sprint Planning meeting within the game. As before, the selection of a simple activity in the form of drawing has multiple benefits: it can be deployed under virtually any conditions, requiring only basic office stationery supplies, and it allows participants to concentrate on experiencing the Scrum development process rather than focusing on game-like work simulation, in many cases involving random elements. Moreover, avoiding random result generation enables a more practical translation of tasks and a more realistic estimation of the time required for their completion.

We conducted the analysis of the game with a group of first-year Applied Computer Science students, who had significant prior theoretical exposure to Scrum through lectures but had not yet had an opportunity to apply Scrum practices in a software development context. This setting has multiple similarities to a day-long workshop introducing Scrum to a team during a Scrum adoption in a company.

This paper is structured into six sections: Section 2 reviews related work on gamification and its application in teaching agile methods, with a particular focus on Scrum; Section 3 describes the details of proposed game and its rules; next Section presents the research methodology; Section 5 presents and discusses the research findings; and Section 6 concludes with a summary of the main insights derived from the study.

2. Related Works

Gamification is the process of incorporating elements from game mechanics into non-game contexts, and it became one of the most actively researched areas around the 2010s [28]. In recent years, gamification has gained widespread popularity as a method for transferring knowledge not only in educational settings (ranging from preschool to university) for teaching core subjects such as mathematics [1], foreign languages, and programming [10], but also in various other areas of life, including promoting engagement and brand loyalty [24], corporate training [19], and health care [3]. Furthermore, the media used to facilitate learning through gamification are diverse, ranging from simple games and web-based testing applications [29] to virtual reality experiences and virtual escape rooms [1]. Most studies confirm the positive outcomes associated with gamification, including increased participant motivation and engagement, as well as measurable improvements in material retention and comprehension [4].

Given the widespread popularity of gamification, it has also become prominent within the field of Computer Science, particularly in teaching soft skills, and in familiarizing learners with methodologies and frameworks, both in software [20] and hardware project contexts [21]. One of the most actively developing areas is the use of games to teach agile methodologies, particularly Scrum and Kanban. Such games are designed to prepare students for the job market [6], [12], [30] as well as to assist companies in transforming existing teams to adapt to new agile approaches [7], [8], [17], [31]. A wide variety of media are employed in these educational processes—from simple card games [6], LEGO-based activities [11], [22], and origami tasks [27] to more technologically advanced solutions, such as Microsoft Excel applications with VBA programming [12], Trello boards [20], Jira Dashboards [16], virtual reality platforms [13], [18], and even video games [25]. Alternative approaches also exist, such as courses utilizing quizzes and story-based gamification [2], [9], or innovative uses of existing games like Minecraft as a task source for teaching Scrum practices [26]. In the case of Kanban, collaborative games are particularly popular, as analyzed in [23]. Many of the Scrum simulation games utilise random elements for work simulations [6], [12] or don't simulate it at all [13], [18], [20], [25], focusing on specific activities such as Story Point estimation, Burndown Chart analysis or just planning multiple Sprints without running them.

However, several examples closely similar to the presented game can be observed (SCRUMIA [30], Lego4Scrum [11], origami game [27]), displaying a practical approach to work simulation without random result generation. All these games, as well as the game presented in this paper, display a similar core stemming from the same goal of teaching the Scrum process. They contain elements presenting in some way or form: different Scrum Roles, Product Backlog, Sprint Planning, Sprint Review, and Sprint Retrospective. However, several key differences can be identified in the specific approaches. Compared to the presented game, contact with the client is less relevant in them. Elements such as undisclosed quality requirements and incomplete user story descriptions are not present. Therefore, increasing the participants' focus on streamlining the development process, without experiencing the elements of requirements refinement and the need for Definition of Done evolution. Other important difference comes in the form of the number of Sprints within a single instance of the game. Only Lego4Scrum, similarly to the presented game, performs at least two full Sprints, allowing for practical implementation of changes based on feedback received after the first Sprint. Origami game and SCRUMIA perform only one full Sprint, focusing mostly on showing basic elements of the Scrum process (e.g. Events and Artifacts) and only discussing the observations during Sprint Review and Retrospective, without the chance to implement changes and observe their results. Additionally, all those games include task estimation and Burndown Chart analysis, while these elements have been purposefully omitted in the presented game, to make client feedback the main source of information regarding teams results and to limit the amount of game elements. Another significant difference is the material used for the work simulation, which, in many cases, even influences the name of the game. In Lego4Scrum, a large number of Lego bricks is required for each team, significantly increasing the cost of preparation and limiting the scalability of the game to multiple teams. Similarly, according to the origami game authors, their game requires special origami paper to ease the complex origami folding process for the participants. In those regards, SCRUMIA presents higher similarity, requiring just old newspapers, similar to basic office supplies used in the presented game.

In summary, the literature offers a wide range of proposals for games designed to teach Scrum practices, encompassing both traditional approaches and those leveraging modern technologies. However, it is noteworthy that one concept largely absent from the existing body of work is the use of a simple, low-complexity task—specifically, drawing basic images or icons—in combination with the complexity of initiallt undisclosed or emerging customer requirements, underlining the importance of empirism and continious improvement. As described in the first limited version of the drawing-based Scrum game [15], the task should be simple enough to allow each participant to focus fully on practicing Scrum elements rather than being challenged by the complexity of the task itself, alowing the participants to better experience the whole empyrical process (transparency-> inspection ->adaptation). A second objective was to ensure that the game could be implemented under virtually any conditions, requiring only basic office supplies. Moreover, the new extended version of the game remains very flexible, despite adding more Scrum Events to the game, Sprint continuity, as well as a full business context to facilitate more realistic Sprint Planning, the game can still be easyli adjusted to focus on specific aspect of Scrum in contrast to most well-known games, which tend to emphasize specific Scrum elements and because of their set structure and required materials can not be easily modified. We aimed to create a solution that could be easily tailored to the needs of different teams, making it suitable both for use in a general Scrum teaching context and for addressing a specific struggle of a given group.

3. Game overview

The game concept is structured around simulating a Sprint within small groups that resemble the Scrum Team. The structure of the simulation and all the required preparations have been presented in Figure 1. At the beginning the participants are organized into groups of three or four individuals, with two participants being assigned the role of Graphic Designers (only role allowed to draw the icons), and the remaining two required roles: Tester and Product Owner are held either by two separate participants (if the total team size is four) or by a single participant fulfilling both roles (if the total team size is three). This group structure ensures flexibility and allows for adaptation to varying initial participant numbers.

Each group is provided with: five markers (two black and three colored), two blocks of post-it notes, a Product Backlog comprising of 15 Product Backlog items (PBI)s, each separate PBI being a different icon, accompanied by a checklist of Acceptance Criteria, a short text outlining the business context (available only to Product Owner, they need to convey the context to the Team), and a table to summarise the course of the simulated Sprint (including fields for Sprint Goal, Sprint Backlog, and the number of PBIs completed by the Team during Sprint and finally accepted by the Customer during Sprint Review). The primary purpose of the Sprint summary table is to capture key information that supports participants during the Retrospective phase and later the summary of the entire learning experience.

After receiving the materials, the participants start the game. The instructor outlines the fundamental rules for the game:

1. Each post-it note must contain exactly one drawn icon.
2. Each icon must meet the established Acceptance Criteria.
3. The colors used for drawing must comply with the specifications provided by the client.
4. Only a Graphic Designer can draw icons

The game begins with the Sprint Planning phase, during which each team familiarizes itself with the business context and the set of PBIs, deciding which PBIs should be completed during the Sprint. The available PBIs consist of icons of different geometrical shapes, animals, people, and everyday objects. The business briefing states that the client would like to release a new expansion in their language learning app featuring geometrical shapes, and the latter planned expansion will probably feature animals. This provides a clear guideline regarding business priorities, but requires the Product Owner to relay this information to the rest of the Scrum Team. At this stage, the instructor should refrain from suggesting specific IPBs or actions to the participants or correcting them, instead allowing them to independently plan their Sprint knowing they will have 3 minutes for drawing and checking icons, only making sure the Teams: define the Sprint Goal, select PBIs for implementation in this Sprint, and write it all down in the Sprint summary table.

Following the Sprint Planning Event, each team is given three minutes to work on their Increment, represented by drawing previously planned icons. The Graphic Designers are responsible for completing the drawing tasks, while the Tester evaluates each icon against the Acceptance Criteria to determine whether it is done or requires improvement. The Product Owner acts as the contact person with the client, addressing any uncertainties raised by the rest of the Scrum Team.

After the three-minute development phase, each team presents its completed icons. A se-

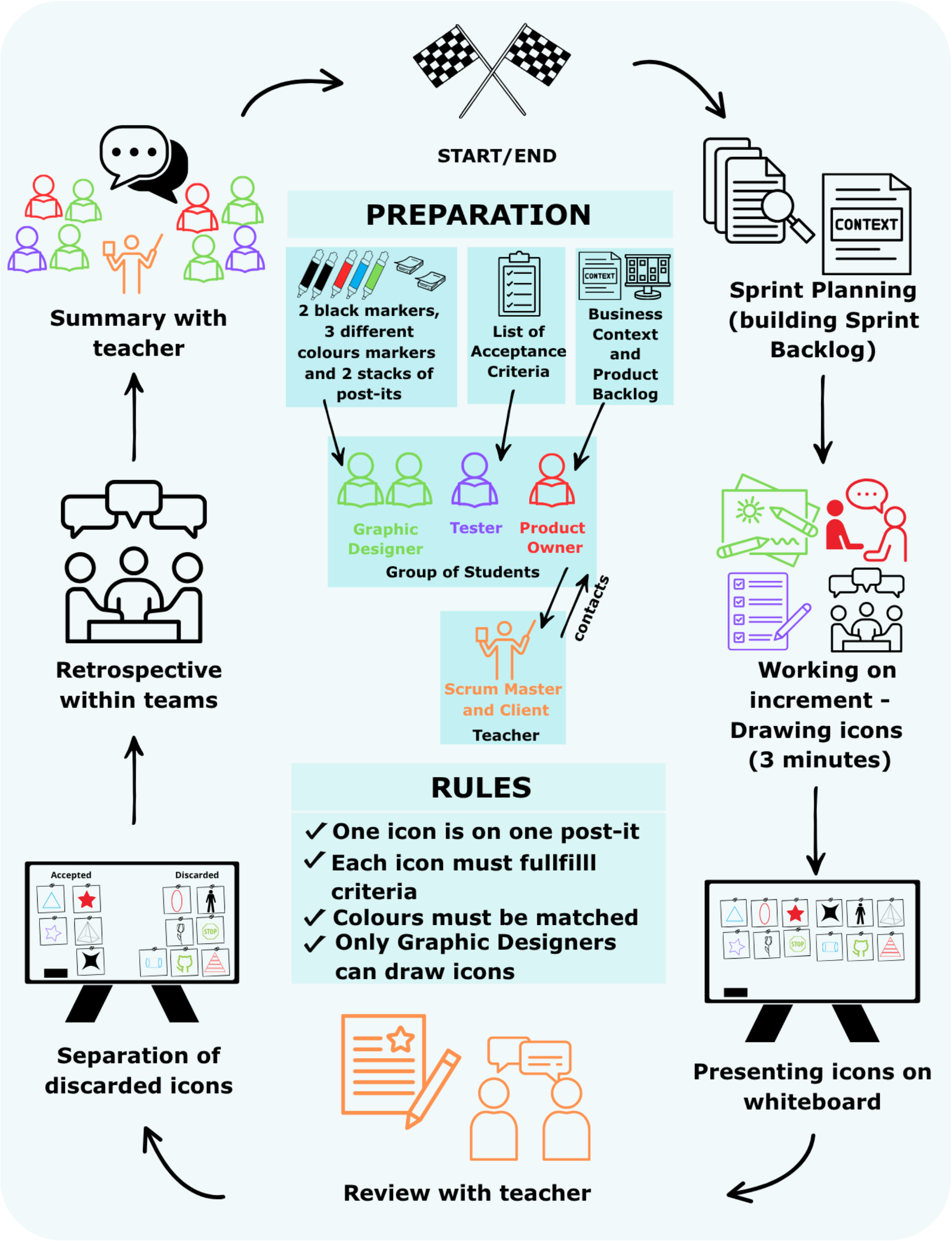
START/END
Sprint Planning (building Sprint Backlog)
CONTEXT
Working on increment - Drawing icons (3 minutes)
Presenting icons on whiteboard
Review with teacher
Separation of discarded icons
Accepted
Discarded
Retrospective within teams
Summary with teacher
PREPARATION
2 black markers, 3 different colours markers and 2 stacks of post-its
List of Acceptance Criteria
Business Context and Product Backlog
Graphic Designer
Tester
Product Owner
Group of Students
contacts
Scrum Master and Client
Teacher
RULES
✓ One icon is on one post-it
✓ Each icon must fullfilll criteria
✓ Colours must be matched
✓ Only Graphic Designers can draw icons

Fig. 1. Overview of game Sprint simulation cycle and required preparation.

lected team member attaches the icons to a shared board. Then, the instructor, acting as the client, reviews the icons and physically separates the accepted icons from those that have been rejected to better visualize the team's results. The instructor assesses the icons against business priorities, the Acceptance Criteria provided, as well as additional implicit criteria not previously specified in the documentation. These previously undisclosed criteria, together with the now-unveiled different understanding of provided requirements, are intended to highlight the importance of frequent inspections and the benefits of an empirical approach to software development and continous improvement by showing participants the benefits of early feedback possible by delivering a small finished increment and performing inspection together with the client.

After the Review, each team performs a Sprint Retrospective, discussing which aspects of their development process were successful and what areas could be improved. Participants also review and, if necessary, update their Definition of Done based on identified new requirements from the client (most common are: where is the top of the icon according to the post-it, how big should the icon be, and how precise should the icon's colouring be). The phase ends with a debriefing session with the instructor, during which teams share their decisions made during Sprint Retrospective. If needed, the instructor can initiate additional discussion, e.g., concerning the initially planned Sprint Goal and discuss it in the context of the already finished Sprint. It gives a good opportunity for participants to notice problems or mistakes in their own approach and to consider experimenting with additional improvements in the next Sprint. After the summary session, the cycle reiterates with a new round of Sprint Planning, where participants can (but don't have to) replan unfinished PBIs, continuing their work with a better understanding of their development process.

The Sprint cycle should be repeated at least twice in total in order to allow participants to experience improvement based on the observations and performed adaptation in the first Sprint. Additionally, it is important to show that during further iterations, new possible improvements are discovered, and new previously inexperienced situations may arise. This shows the importance of the continuous improvement approach and alerts participants not to skip events such as Review or Retrospective after a few Sprints in the future.

4. Methodology

In order to evaluate the effectiveness of the game in teaching Scrum, a Scrum knowledge assessment test and a post-session survey were prepared. Participants completed one knowledge assessment test before and another after the game, while the survey was administered at the end of the session to capture participants' perceptions and experiences. The knowledge assessment test comprised eight closed-ended, single-choice questions, designed to evaluate participants' understanding of more advanced topics related to Scrum (Sprint Planning - Q1 & Q4, the Sprint Backlog - Q3 & Q6, testing in Scrum - Q2, the Definition of Done - Q5, and the Sprint Retrospective - Q7 & Q8). Each question also included the option "I don't know" to capture uncertainty and prevent guessing. The difference in answers between the pre-session and post-session tests aimed to assess whether the use of the game's individual components had successfully enhanced participants' familiarity with different aspects of Scrum, thus evaluating the effectiveness of the game in achieving its educational objectives.

The post-session survey that took place after the game consisted of 6 statements regarding participants' feelings after the same, and a single open question. For the 6 statements shown in Table 1, participants used Likert scale ("Strongly agree," "Somewhat agree," "Neither agree or disagree," "Somewhat disagree,", "Strongly disagree." or "I don't know") to express if they agree with the stated opinion. The first three statements evaluated participants' sense of familiarity with the discussed concepts, and the fourth statement considered their feeling of readiness to work within the Scrum framework. Additionally, the survey included a statement to assess the effectiveness of the chosen knowledge transfer method, specifically, the use of drawing as a parallel for performing programming tasks, compared to other approaches, such as the random generation of results (statement five). The final statement focused on participants' perceptions of the overall form of the game, specifically whether it was enjoyable and engaging. Furthermore, participants were given the opportunity to provide open-ended comments, enabling the authors to gather qualitative feedback for refining and enhancing the game in future iterations.

The study was conducted with 107 first-year students of Applied Computer Science at the Warsaw University of Technology. During the first semester, students were introduced to the fundamentals of Scrum through lectures, while in the second semester, they were required to complete a project utilizing Scrum practices. Before starting their project work, the students

Table 1. Statements used in the post-session survey.

	Survey statements
S1	After playing the game, I better understood the difference between a rigid (e.g. waterfall) and agile empirical approach to software development.
S2	After playing the game, I better understood how we ensure quality in Scrum.
S3	After playing the game, I better understood when it is worth using Scrum.
S4	After playing the game, I feel more ready to work in Scrum.
S5	Drawing as specific tasks in a Sprint helped me understand the Scrum principles better than randomly generated results.
S6	The drawing option in the game was interesting.

engaged in a described game designed to help with building a practical understanding of Scrum and its empirical mechanisms. This situation represents a common scenario for teaching Scrum both in university education and in a common one or two-day workshop training in industry, where most of the knowledge is being taught by the instructor. Although participants may possess theoretical knowledge of Scrum, they often lack practical experience in applying its individual elements as expected in a professional environment.

Students participating in the game were divided into subgroups according to the game's preparation guidelines. Before starting the game, they completed a knowledge test on Scrum. Subsequently, the teams received instructions regarding how the simulation works. After the introduction, the game started and always included two full cycles. After completion of the game, all participants were required to complete the knowledge test once more, along with a survey. All assessments were administered in paper form during the class sessions, immediately before and after the gameplay. Following the collection of tests and surveys from all participants, the responses were systematically tabulated for further analysis. No partial answer was submitted. During the answers inspection, no obviously biased or fake answer was detected. Therefore, no answer was deleted or omitted.

5. Results and discussion

Within each class, participants were able to complete two full Sprints of the simulation. Based on instructors observations, after the first Sprint, most participants demonstrated an ability to identify gaps in their processes, revise or enhance their Definition of Done, and improve their development process within teams (e.g., dividing materials, setting policies regarding clarifying requirements with the Tester, and better organizing the order of icon creation based on business value). Because the results of both Sprints were posted on a whiteboard, participants could easily observe their higher productivity in the second Sprint, coming from improvements introduced after the first Sprint. With few exceptions, participants showed high levels of engagement, readily taking on subsequent tasks and actively discussing both their successes and areas for improvement. After completing the entire game, participants were easily able to draw analogies between the exercise and the software project they were expected to complete during their course. Observations from the post-game survey responses also indicate that participants generally perceived the game very positively, as shown in Figure 2.

Regarding Statement 1, which addressed understanding the differences between rigid and agile empirical approaches, over half of the participants (63%) strongly agreed that the game helped them comprehend these differences, while 35% somewhat agreed. Only 2% of participants remained neutral. No negative responses were observed. These results suggest that directly translating Scrum principles into a physical exercise enabled participants to better perceive analogies with the software development process.

Responses to Statement 2, which concerned quality assurance in Scrum, were similarly

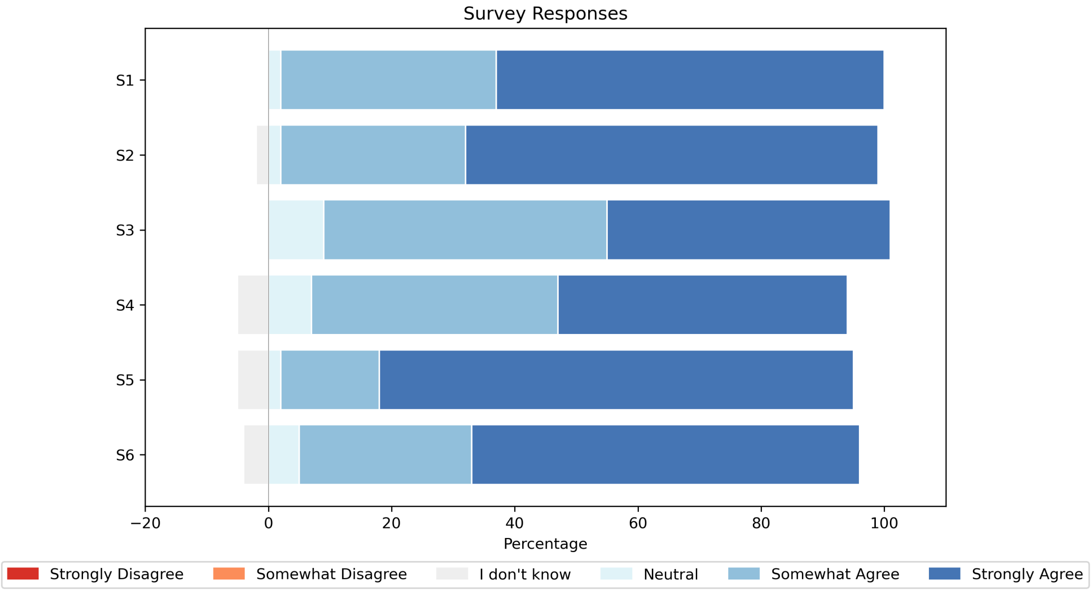

Fig. 2. Post-session Survey responses.

positive: 66% of participants strongly agreed that the game improved their understanding of this concept, 30% somewhat agreed, and 2% remained neutral or selected "I don't know." These outcomes were reflected in the participants' behavior during the Retrospectives, where they were able to update the Definition of Done and discuss strategies to ensure quality internally within their teams, which resulted in significantly better results in the second iteration.

For the statement concerning the ability to recognize situations in which an agile approach is appropriate (Statement 3), 46% of participants strongly agreed and another 46% somewhat agreed, while 9% remained neutral. In the subsequent statements, responses were slightly more varied. A total of 88% of participants indicated that after playing the game they felt ready to work in Scrum (47% strongly agreed and 41% agreed). However, 7% remained neutral and 5% selected "I don't know".

Notably, 93% of participants agreed (77% strongly agreed) with the statement that drawing served as a better analogy for Scrum tasks than random result generation. Only 2% remained neutral, and 5% were unsure how to respond. Additionally, 91% of participants found the drawing activity interesting, while 5% were neutral and 4% did not respond. These findings suggest that selecting a simple, concrete task, such as drawing, significantly enhances the construction of analogies between gameplay and real-world software development processes, as opposed to using virtual randomization. Additionally, participants also exhibited higher levels of engagement, having a physical and creative activity to perform.

Approximately 22% of participants (24 individuals) provided open-ended comments. Many emphasized their appreciation for the class format, noting that it helped them understand the agile approach in practice. Participants valued the opportunity to address gaps in their knowledge through practical exercises that did not require immediate focus on programming skills. Performing simple tasks, such as drawing icons, enabled them to fully experience the Scrum process and seek clarification when needed. Some participants suggested that the Product Owner role could be made more dynamic by introducing additional issues requiring negotiation with the client. Several participants also noted that the time allocated for drawing icons was too short.

The knowledge test results indicate that participation in the game significantly increased participants' awareness and understanding of Scrum. For every question, an improvement in the number of correct responses was observed as shown in Figure 3.

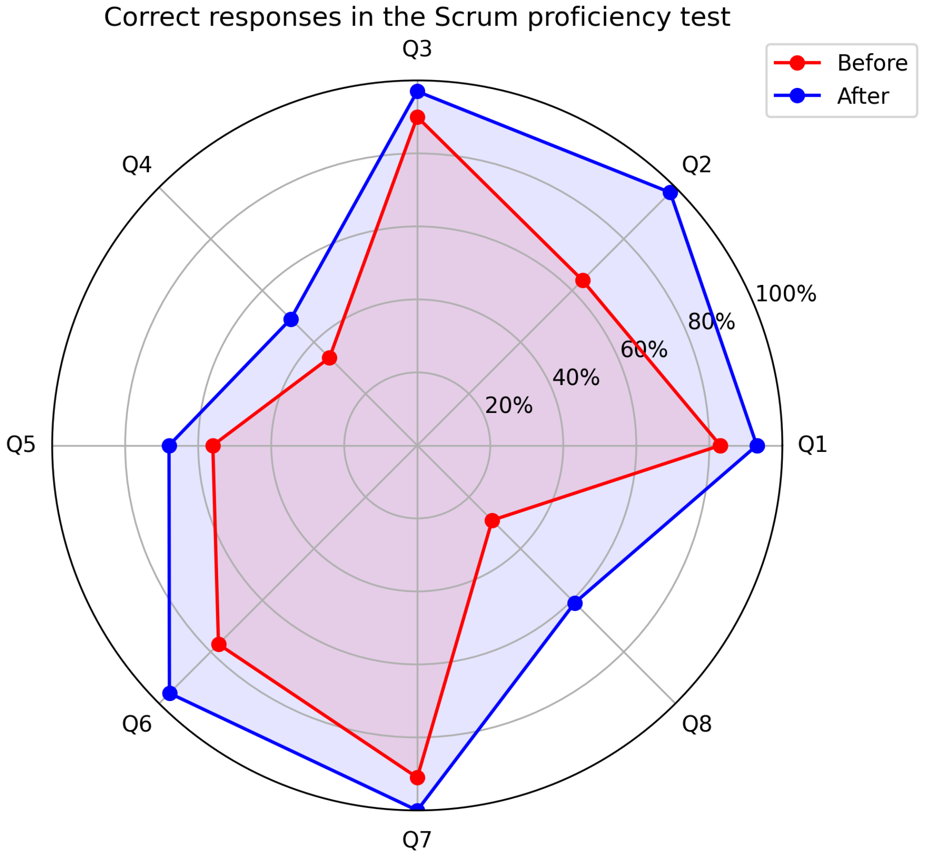

Fig. 3. Scrum knowledge assessment test results before and after the simulation.

Questions 1, 3, and 7 were relatively easy for participants from the outset—prior to playing the game, 83%, 90%, and 91% of participants, respectively, answered these questions correctly. However, it is noteworthy that after the game session, the proportion of correct answers further increased to 93%, 98%, and 100%, respectively. This suggests that the game successfully reinforced participants' understanding of key Scrum concepts related to Sprint Planning, the preparation of the Product Backlog, and Retrospectives. These elements are naturally emphasized during the gameplay. In this regard, the game not only enhanced comprehension of fundamental Scrum practices but also contributed to team integration, promoting appropriate practices and fostering cooperation and self-organization.

The remaining questions were initially more challenging for participants. Questions 2, 6, and 8, which addressed the topics of testing, prioritization of work within a Sprint, and ending the Sprint, respectively, show a significant rise in correct answers. Specifically, for Question 2, the proportion of correct answers increased from 64% before the game to 98% after the game. For Question 6, the correct response rate rose from 77% to 100%, and for Question 8, it improved by 32 percentage points (from 29% to 61%). These substantial gains confirm the effectiveness of the game as a teaching tool for Scrum practices.

For Questions 4 and 5, the improvement in correct answers was more modest, approximately 12–15% (an increase from 34% to 49% for Question 4, and from 56% to 68% for Question 5). This shows that topics regarding the evolution of the Definition of Done and possible reworks are not sufficiently emphasized in the new, extended version of the game. A more explicit distinction between increment testing within the Scrum Team and final client feedback should be considered in the future versions. Nevertheless, these results suggest that even without explicit instructional focus on those particular practices, the game itself contributed to improvement in participants' knowledge of Scrum in those areas. It is important to note that to maintain consistent performance through the game evaluation, the results of the initial survey weren't considered by the instructor during the simulation. However, due to the open structure of the game, such an adjustment to a specific scenario could be done without previous preparation. This remains a subject for future study.

Additional positive results of the game have been observed later by the teachers during the students' project using Scrum that followed the workshops during which the experiment have been performed. Both teachers observed better then usuall results, especially in the area of Scrum knowledge and its practical aplication during teams' software development part of the classes. Additionally, it was observed, that the analogy between gameplay (icon drawing) and real software development was wery usefull while corecting students mistake and errors, by providing a reference point, fully understandable by the students. Howevery, it is crucial to note, that those observations were purely subjective, as those aspects weren't part of the studies methodology and should be treated as additional observations and inspiration for future studies.

6. Conclusion

In this paper, we introduce an extended version of a drawing-based game designed to simulate work within a Scrum framework. The primary objective of the game is to demonstrate Scrum practices in a realistic work environment by means of an analogy involving the drawing of icons. Games complex simulation of Scrum Events allows participants to practically apply empiricism to improve their development process and observe the results of their improvements on their work. Because no randomly generated results are used during the simulation, participants can't blame "being unlucky" as a cause for bad results. Therefore, immediately focusing on possible improvements to their development process and allowing for building an easy parallel between the game experience and the Scrum development process.

We conducted an empirical study involving 107 first-year Applied Computer Science students, who had previously acquired theoretical knowledge of Scrum and, in most cases, didn't have real-life experience working in Scrum. The analysis revealed that the one-hour game session significantly improves participants' knowledge of more advanced aspects of the framework, as shown in a noticeable increase in the number of positive answers to every question in the knowledge test. This has been achieved, despite participants already having a good theoretical knowledge of Scrum prior to the game with almost all questions have over 50% correct answers at the start. Furthermore, the participants evaluated the game positively, describing it as both engaging (91%) and educational (98% and 96%). Additionally, participants agreed (93%) that drawing icons served as a better means of simulating work rather than randomly generating results, a method commonly used in other Scrum simulation games.

We contend that our experimental evaluation successfully demonstrated the feasibility of using the game to reinforce Scrum practices and validated our initial hypotheses. Nevertheless, further studies are warranted, particularly within entrepreneurial environments. Game validation on more diversified teams, comprising both experienced and novice Scrum practitioners, where harmonization of practices and establishment of a shared mission would be necessary, could provide valuable insights for enhancing the game's utility in coaching contexts. Additionally, the validation of game adjustment based on initial Scrum knowledge assessment remains an important area for future studies.

References

[1] Baziak, R., Daruk, T., Żyra, K., Żarek, D., and Lebiedź, J.: Virtual Escape Room in Mathematics. In: *Harnessing Opportunities: Reshaping ISD in the post-COVID-19 and Generative AI Era (ISD2024 Proceedings)*. 2024.

[2] Ciucă, G., Ciupe, A., and Orza, B.: Exploring educational scenarios through interactive environments and agile user stories: a gamified assessment case study. In: *2022 International Symposium on Electronics and Telecommunications (ISETC)*. IEEE. 2022, pp. 1–4.

[3] Damaševičius, R., Maskeliūnas, R., and Blažauskas, T.: Serious games and gamification in healthcare: a meta-review. In: *Information* 14.2 (2023), p. 105.

[4] De Freitas, S.: Are games effective learning tools? A review of educational games. In: *Journal of Educational Technology & Society* 21.2 (2018), pp. 74–84.

[5] Digital.ai: *16th Annual State of Agile Report*. Accessed: 2025-05-01. 2022. URL: `https://stateofagile.com/`.

[6] Fernandes, J. M. and Sousa, S. M.: Playscrum-a card game to learn the scrum agile method. In: *2010 Second International Conference on Games and Virtual Worlds for Serious Applications*. IEEE. 2010, pp. 52–59.

[7] Hermanto, S., Kaburuan, E. R., and Legowo, N.: Gamified SCRUM design in software development projects. In: *2018 international conference on Orange technologies (ICOT)*. IEEE. 2018, pp. 1–8.

[8] Imron, A. S., Raharjo, T., Hardian, B., Simanungkalit, T., JULIANASARI, R., RAHARJO, T., HARDIAN, B., SIMANUNGKALIT, T., TAGHZOUTI, A. O. A., BOUDALLAA, I., et al.: GAMIFICATION TO IMPROVE SCRUM ADOPTION: A CASE STUDY AT POULTRY STARTUP IN INDONESIA. In: *Journal of Theoretical and Applied Information Technology* 100.20 (2022), pp. 5854–5864.

[9] John, I. and Fertig, T.: Gamification for software engineering students-an experience report. In: *2022 IEEE Global Engineering Education Conference (EDUCON)*. IEEE. 2022, pp. 1942–1947.

[10] Kasenides, N. and Paspallis, N.: aMazeChallenge: An interactive multiplayer game for learning to code. In: *29th International Conference on Information Systems Development*. Association for Computing Machinery (ACM). 2021.

[11] Krivitsky, A.: lego4scrum: A complete guide. A great way to teach the Scrum framework and Agile thinking. Amazon, 2020.

[12] Lee, W. L.: SCRUM-X: An interactive and experiential learning platform for teaching scrum. In: *The 7th International Conference on Education, Training and Informatics (ICETI 2016)*. Elsevier, 2016.

[13] López-Fernández, D., Mayor, J., Pérez, J., and Gordillo, A.: Learning and motivational impact of using a virtual reality serious video game to learn scrum. In: *IEEE Transactions on Games* 15.3 (2022), pp. 430–439.

[14] Lynch, D.: Improving teaching through coaching, mentoring and feedback: A review of literature. In: *MIER Journal of Educational Studies Trends and Practices* (2014), pp. 136–166.

[15] Marek, K. and Martyniuk-Sienkiewicz, K.: Drawing Based Game for Teaching Scrum. In: *International Conference on Agile Software Development*. Springer. 2024, pp. 203–208.

[16] Marques, R., Costa, G., Mira da Silva, M., Gonçalves, D., and Gonçalves, P.: A gamification solution for improving Scrum adoption. In: *Empirical Software Engineering* 25.4 (2020), pp. 2583–2629.

[17] Marques, R., Costa, G., Silva, M. M. da, Gonçalves, D., and Gonçalves, P.: Using Gamification for adopting scrum in practice. In: *27th International Conference on Information Systems Development: Designing Digitalization, ISD 2018*. Association for Information Systems. 2018.

[18] Mayor, J. and López-Fernández, D.: Scrum vr: Virtual reality serious video game to learn scrum. In: *Applied Sciences* 11.19 (2021), p. 9015.

[19] Murwonugroho, W.: Creative gamification in kahoot! For worker's health and safety learning assessment. In: *International Journal of Scientific and Technology Research* 9.3 (2020), pp. 1992–1998.

[20] Naik, N. and Jenkins, P.: Relax, it'sa game: Utilising gamification in learning agile scrum software development. In: *2019 IEEE Conference on Games (CoG)*. IEEE. 2019, pp. 1–4.

[21] Omidvarkarjan, D., Hofelich, M., Conrad, J., Klahn, C., and Meboldt, M.: Teaching agile hardware development with an open-source engineering simulator: An evaluation with industry participants. In: *Computer applications in engineering education* 31.4 (2023), pp. 946–962.

[22] Paasivaara, M., Heikkilä, V., Lassenius, C., and Toivola, T.: Teaching students scrum using LEGO blocks. In: *Companion Proceedings of the 36th International Conference on Software Engineering*. 2014, pp. 382–391.

[23] Przybylek, A. and Olszewski, M. K.: Adopting collaborative games into Open Kanban. In: *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE. 2016, pp. 1539–1543.

[24] Rodrigues, I., Lopes, J. M., Borges, A., Oliveira, J., and Oliveira, M.: How can Gamified applications drive engagement and brand attitude? The case of Nike run club application. In: *Administrative Sciences* 11.3 (2021), p. 92.

[25] Rodriguez, G., Soria, Á., and Campo, M.: Virtual scrum: A teaching aid to introduce undergraduate software engineering students to scrum. In: *Computer Applications in Engineering Education* 23.1 (2015), pp. 147–156.

[26] Schäfer, U.: Training scrum with gamification: Lessons learned after two teaching periods. In: *2017 IEEE Global Engineering Education Conference (EDUCON)*. IEEE. 2017, pp. 754–761.

[27] Sibona, C., Pourreza, S., and Hill, S.: Origami: An active learning exercise for scrum project management. In: *Journal of Information Systems Education* 29.2 (2018), pp. 105–116.

[28] Swacha, J.: State of research on gamification in education: A bibliometric survey. In: *Education Sciences* 11.2 (2021), p. 69.

[29] Swacha, J. and Kulpa, A.: A Game-Like Online Student Assessment System. In: *Harnessing Opportunities: Reshaping ISD in the post-COVID-19 and Generative AI Era (ISD2024 Proceedings)*. 2024.

[30] Von Wangenheim, C. G., Savi, R., and Borgatto, A. F.: SCRUMIA—An educational game for teaching SCRUM in computing courses. In: *Journal of Systems and Software* 86.10 (2013), pp. 2675–2687.

[31] Yilmaz, M. and O'Connor, R. V.: A Scrumban integrated gamification approach to guide software process improvement: a Turkish case study. In: *Tehnički vjesnik* 23.1 (2016), pp. 237–245.