# Learning Process of Agile Scrum Methodology with Lego Blocks in Interactive Academic Games: Viewpoint of Students

Douglas Augusto Barcelos Bica and Carlos Alexandre Gouvea da Silva, *Member IEEE*

***Abstract*—The rapid growth of Information Technology led to the development of a multitude of smartphone systems worldwide. As a result, the number of educational institutions offering courses in areas such as programming and software engineering increased. However, traditional processes for software development did not keep pace with changing technologies. In the last few years, software development became more dynamic and iterative, requiring stakeholders to work as a team and deliver higher quality projects in less time, by using methods such as Agile Development (e.g., Scrum and Extreme Programming (XP)). Although some institutions approach this content in graduation courses, many students and professor are indifferent towards it, resulting in low enthusiasm and practice. This article presents the real case of a classroom activity to teach Scrum concepts by using Lego blocks. At the end of the classes, students were asked to rate the effectiveness of the activity. The results showed that dynamic games and palpable activities are more effective than theoretical or video lessons.**

***Index Terms*—Software development, Agile Process, Scrum, Learning Education**

## I. INTRODUCTION

IN the last few years, software development has been defined as a set of manufacturing activities [1]. Traditionally, the process was thought as being a series of interrelated activities [2], which entailed definition of project steps, feasibility analysis, requirements engineering, design, coding, verification tests and validation. In previous literature, classic processes such as Waterfall and V-shaped models did not reflect the iterative nature of exploratory development.

Traditional processes are historically important in software development, nevertheless, iterative and evolutionary Agile Development methodologies have replaced them. Agile Development uses additive and iterative development, by repeating numerous phases within a process and improving the software by auditing customer feedback that converges in new solutions [2]. The main Agile methodologies used today are Scrum and Extreme Programming (XP) [3].

Agile Development is accepted by most technology companies, nevertheless, it is a challenging methodology to carry out since it tends to suffer resistance from all organizational levels [4]. Nerus et al. [5] indicate that four key issues tend to encumber the migration to Agile Development: 1) Management and organizational culture, style and rewards system; 2) Employees team relationship and technical competence; 3) The change from a process-centric development to a feature-driven and iterative one; 4) The existence of suitable technologies and tools. Scharff et al. [6] show that most development issues are related to the different developing pace and the distance between stakeholders and culture. Still, Agile methodologies are being accepted among traditional organizations, most of them indicating a preference for maintaining both styles of development [7].

Nowadays, professionals must be prepared to face challenges within companies. In undergraduate courses, Scrum and XP are widely used in classes and both approaches focus on meeting the student learning needs [8]. Teaching Agile methodologies in educational institutions tends to be more effective if done in an agile environment: valuing students that fit the method best, building their skills through real practice [8]. Scrum is an Agile Methodology which focuses on project management and allows iterative development. However, the learning process of Scrum does not tend to be effective, due to inadequate learning environments and inexperienced tutors that lead to unmotivated students. Considering this, many institutions have been using active methodologies to improve the learning and teaching process [9][10].

Traditionally, classical learning process is based on two elements: professors and students. In this case, the professor transmits his/her knowledge to a class while students listen to the information, make notes or questions reply. This learning process allows acquisition of new mental schemes, knowledge, skills, abilities, and among other, in which could be used to solve elementary or complex problems, making use of tools, simulators, mathematical equations, theories or analytical analyzes. Among the years, the learning process evolved in different ways. Felder and Silverman [11] described this process as a structure of two-steps in which firstly the students receive information and then they process what must keep memorized. For Prince and Felder [12] the teaching and learning process is based on a set of a variety of instructional methods, including question-based learning, problem-based learning, project-based learning, case-based teaching, discovery learning, and just-in-time teaching. Armstrong and

Manuscript received July 19, 2019; revised November 12, 2019 and March 5, 2020; accepted March 18, 2020; date of publication DATE; date of current version DATE. (Portuguese version received DATE; revised DATE; accepted DATE.

D. A. B. Bica was with the Department of Information System and Agile Development, University of Araucária (UNIFACEAR), Araucária City 83707-067, Paraná, Brazil; e-mail:dougbrasil.bica@gmail.com.

C. A. G. Silva was with the Department of Information System and Agile Development, University of Araucária (UNIFACEAR), Araucária City 83707-067, Paraná, Brazil; carlos.gouvea@ieee.org

There exists a Portuguese version of this article available at http://rita.det.uvigo.es/VAEPRITA/V8N1/A.pdf Digital Object Identifier Pending DOI:

Fukami [13] define learning as a holistic process of adaptation, in which involves the total integrated functions of people, such as thinking, feeling, perceiving and behaving for problem-solving, decision-making and creativity.

Nowadays, several methods to teach Scrum are presented, such as games [14][15], simulators, virtual software [16][17], video lessons, lectures, among others. For some works, i.e., [18][19][20], the Scrum is used to support the development of real software projects, which can improve not only the productivity of software implementation but also the understanding of students about the Scrum. However, these works did not present the students perspective and their feelings about learning.

We propose the analysis of a dynamic Scrum activity based in Lego blocks, which was applied in undergraduate classes. The main innovative value of this paper is to contribute with the adaptation of an interesting game for students to understand and apply Scrum's main concepts, learning to work as a team, with responsibilities and time management. At the end of the experiment, we asked students how effective this method was in comparison to other ones they experienced. Based on the feedback of students it is expected they increased their knowledge about Scrum topics, such as process steps of Scrum, main roles, and artifacts. When compared to other traditional learning processes it is expected from students of this activity better feelings and good motivations.

In Section II, the Scrum methodology is presented. Section III briefly presents some of the techniques and methods used to teach Scrum in educational environments. Section IV explains the game used to teach Scrum and how it improves students learning. In section V, results are shown and some arguments are raised. In Section VI, conclusions and ideas for future works are presented.

## II. SCRUM FRAMEWORK

The origin of Scrum can be traced to the article "The new new product development game" published in 1986 by Takeushi and Nonaka in Harvard Business Review [21]. In the article, authors use an analogy that compared ruby games and product development teams. In 1993, Jeff Sutherland and Ken Schwabe applied the concepts suggest by Takeushi and Nonaka in Easel Corp business environment. The application of these concepts came to be known as Scrum. Initially, it started as the development of commercial products within in a 30 day iteration process [22]. Yet, the use of Scrum for software development was not considered.

In 1995, Ken Schwaber published an article with the title "Scrum development process", in which he introduced Scrum concepts for software development [23]. Since then, several adjustments have been made to the methodology but the fundamentals are still the same. Scrum can be described as a framework which looks at the development process as flexible set of activities, combining tools and techniques to make development teams achieve the best results they can in building systems. Moreover, Scrum is an incremental development cycle, composed of periodic short-time deliveries [24] which uses iterations in order to obtain the final software product. In Figure 1, an overview of the Scrum process is shown.

Within the Scrum Framework, three roles are defined: Product Owner (PO); Development Team (DT) or Scrum team; and Scrum Master (SM). The main activities of each stakeholder of Scrum are presented in Table I.

TABLE I
RESPONSIBILITIES AND ACTIVITIES OF PO, DT AND SM [20]

| Stakeholder | Activities |
|---|---|
| PO | The PO obtains the initial and on-going funding to carry out the project, creating the initial functional and non-functional requirements of the system and the expected investment objectives. The list of requirements is called Product Backlog (PB) in which the PO uses to define which features have the highest value and which should be developed initially. The PB is continuously updated by the PO while system is constructed within iterations. |
| DT | DT is responsible to develop functionalities from the PO priority list based on PB. The team develops PB based in functionality increments within iterations using the technical knowledge of all stakeholders. |
| SM | SM is responsible for Scrum process, for teaching all the rules and practices of the methodology, assists in the elimination of difficulties and impediments that can harm the project, and facilitates important meetings and decisions. |

Initially, the PO shows the team a list with all the functionalities that must be developed. These functionalities are known as User Stories (US). One of the most difficult parts of the process is to define how much effort will be required to develop each US. In order to solve this issue, a wide range of techniques is used. Most Agile methodologies recommend the use of Planning Poker, as a technique to calculate an estimative of the team size necessary to develop the required US [25]. In most cases, the definition of required effort is defined by the DT, by choosing an Effort Value based in Fibonacci numerical sequence (1, 2, 3, 5, 8,..., maximum) [26]. A lowest Fibonacci value represents a low effort to build an US and a maximum value represents the highest effort. Typically, a number ranging from 0.5 up to 20 is used to describe the effort of each US [27], but other sequences can be used as well. In some situations, especially in small teams, it is common to define each effort in working hours [28].

Scrum divides projects into Sprints (iterations), but before starting a Sprint, it is required to execute a Sprint Planning (SP) that is essential to ensure the project success [29]. A Sprint Planning is a meeting in which the PO indicates which US are most crucial ad which should be developed in the next Sprint. In this step, the DT only considers US that are capable of being executed in the next Sprint, avoiding the ones that require more effort than what is available. This meeting will result in a Sprint Backlog (SB). The SB is a list of US which the DT proposes to complete in the Sprint [30].

During a Sprint, the DT develops all necessary activities and actions to build the US, according to PO's requirements. During this phase, the SB should not change. Due to the need for communication between stakeholders, a daily meeting is performed [31]. This Daily Scrum meeting takes approximately 15 minutes, in which the team has the opportunity to share their work with each other and report any existing obstacle. Usually, team members report one of the three information:

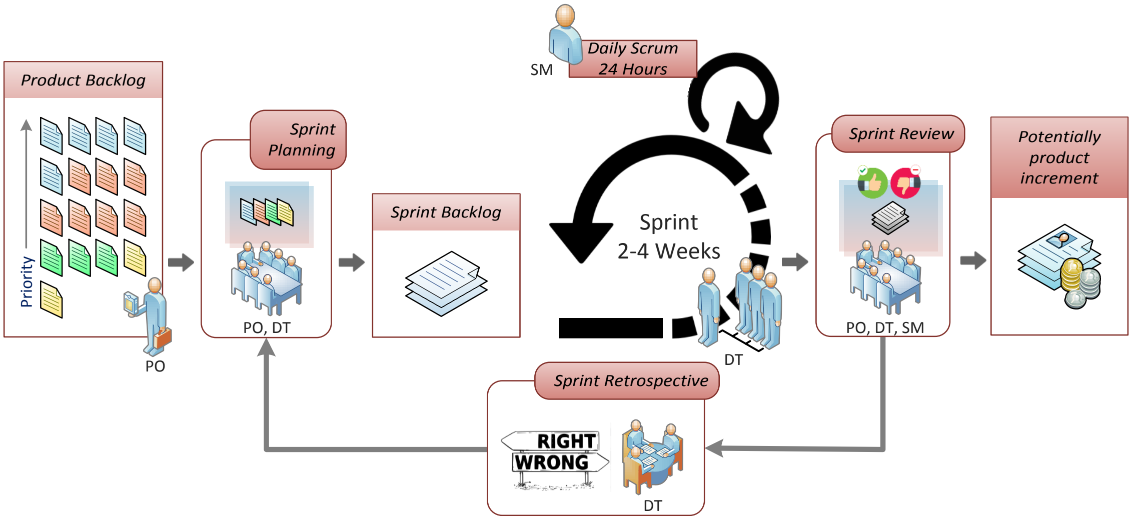


Fig. 1. Sequence of steps based in Scrum Framework.

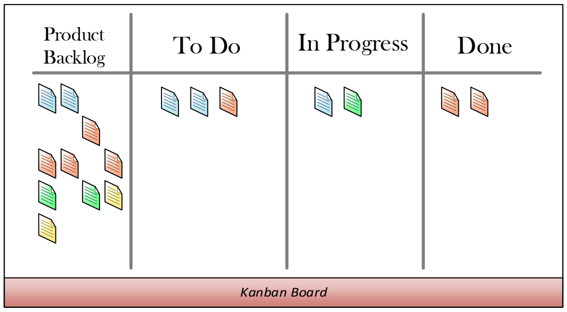


Fig. 2. Representation of a classical Kanban board.

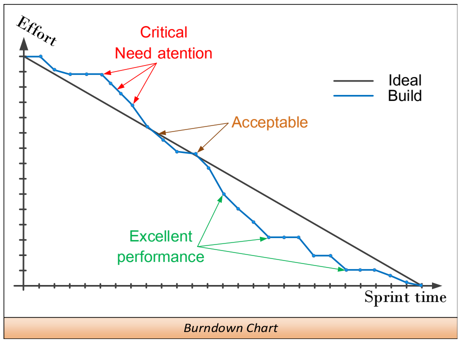


Fig. 3. Burndown chart with ideal line vision and performance sprint performed.

- What they were able to get done since the last meeting;
- What they are planning to finish by the next meeting;
- Any impediments that are in their way.

During Daily Scrum meetings, the team members update two tools that allow easy communication and visualization of the development process, i.e Kanban Board and Burndown chart. Kanban is a tool which has the goal to keep a board of the US that are under development in the current sprint. The most common model is divided into four columns: “To do”, “In progress”, “Done” and the Backlog, as shown in Figure 2. US which are still not started are put in the “To Do” column, the ones in development are put in the “In progress column” and the finished US are put in the “Done” column. During the whole process, the US are moved between each column, allowing a better understanding of the team’s performance according to the situation of each US.

In Daily Scrum, the DT updates the Burndown chart. The Burndown chart shows each day a new estimate of how long (effort or hours) it lasts until all US are finished. The first day of a Sprint is presented by a value that indicates the sum of all efforts estimated during the planning meeting, this value is expected to be zero on the last day of Sprint. Therefore, a “ideal” line of maximum Effort value up to zero is drawn, as shown in Figure 3. Each day, the remaining effort is estimated and indicated on the Burndown chart. If the remaining Effort is above the expected value, then performance is poor, if the contrary, performance is considered excellent.

At the end of a Sprint, the US are delivered. This step is called the Sprint Review (SRv) and consists of showcasing the PO what was developed. The PO analyzes whether the delivered product corresponds to what was requested, providing a level of quality control. Eventually, the US that were not developed or were not considered optimal by the PO return in a future Sprint.

After the SRv, a Sprint Retrospective (SRt) occurs at the end of every Sprint. At this stage, the DT analyzes the performance of team members and indicate how can they improve in future Sprints. During this stage, it is common for a team to question aspects such as:

- What worked or went well?
- What caused problems, failed to work properly, or did not go well?
- What can they do differently in the next Sprint to improve performance in order to avoid issues?

Once the SRv and SRt are finished, the PO can update the PB with modifications or new US if necessary. From this step, the DT can start another Sprint Cycle based in a new SP meeting.

## III. RELATED WORKS

The learning process can be affected by several factors, including low motivation, teaching methods and classroom tools. The use of practical activities are an interesting alternative to the process of learning Scrum. In university, students learn to develop skills and attitudes using practical activities [32]. In this section, we show a brief overview of works that used different methods to learn Scrum in higher education.

Lee [16] suggests the use of Scrum-X, a software that help students boost their understanding of the Scrum methodology. Scrum-X is a simulation platform used to improve Scrum concepts through experimental exercises. The goal is managing a software project, by developing a book sales website, encompassing almost all Scrum concepts. The learning impact of this software was showcased in a project with 15 students divided into three groups. At the end, a questionnaire was used to evaluate the experiment effectiveness. Results indicate a positive potential and the authors report a positive impact in student's motivation to learn and engagement.

Kropp et al. [33] suggested a Scrum activity using Lego blocks, in which students have to build a city. The construction must be carried out in 4 sprints of 25 minutes each. The authors observed that the teams start with a weak Sprint, but through experiment they reached an optimal speed. Results shown report that students liked the practical approach, appreciating the intense communication and collaboration of team members. Most students tended to approve documents in meeting very quickly and focused on customer delivery, although they recognized the importance of writing the US, a task they considered difficult but useful. Steghofer et al. [34] presented results of an experiment with 450 students participating in Scrum Workshops that used Lego blocks to build a city in multiple short Sprints.

Having the objective of evaluating the impact of practical activities, Castro et al. [35] used Scrum to simulate the development of simple electronic components. The Kanban board was improved by the SM to follow the stories developed by the team, with each member doing sprints of 6 minutes each. At the end of each sprint, a card was withdrawn from a deck of penalties or benefits, adding dynamism to the simulation. An evaluative questionnaire with grades raging from 1 to 5 was applied in order to quantify the impact of the experiment on student's motivation, in comparison to theoretical teaching. According to the author, better learning, motivation and student satisfaction was reached, although students faced interpersonal difficulties.

Games have been gaining visibility as Educational Tools [36][37][38]. Using Lego Blocks, Paasivaara et al. [39] used a game to teach Scrum in a Finnish university. During the game, students became familiar with Scrum roles, steps, events and concepts practice, by simulating several Sprints, including planning and building a product out of Lego blocks. Results show that students had a high learning and satisfaction levels.

Most of works in this section presented results regarding the motivation and interest of students in activities that use Lego software or blocks to teach Scrum. Although students were given questionnaires to evaluate these results, they were not asked whether this activity is better for teaching Scrum than other traditional or classroom activities. The main contribution of this paper is also to present this perspective of the students.

## IV. MATERIALS AND METHODS

In order to teach Scrum to undergraduate students, we organized Workshops using Lego blocks and invited them to participate. Three workshops were organized in two different universities in Brazil. Workshops information is shown in Table II.

TABLE II
INFORMATIONS ABOUT WORKSHOPS

| Workshop | University | Undergraduate Course | Number of Students |
|---|---|---|---|
| 1st | X | Information System | 14 |
| 2nd | Y | Electrical Engineering | 20 |
| 3rd | X | Information System | 27 |

Each workshop introduced Scrum to freshmen and students who did not know the method or how it worked. First-year students were invited as well, to allow the understanding of the learning process in different knowledge levels.

After an introduction to Scrum, we presented the challenge of building a city using Lego Blocks by using the methodology. The sequence of steps performed in this workshop is shown in Figure 4. The Lego City game consisted of four sprints, including the execution time of each step.

In order to perform the activity, three types of stakeholders were defined: a PO, who was the workshop operator, was responsible for keeping the PB up to date; a SM was appointed to collaborate with the DT and keep every team member involved in the Scrum methodology; and finally, the participating students carried out the role of being DTs.

Students were divided into groups of 5 or 6. Each group had to build different parts of the city, based in a list of requirements made by the PB. First, the PB presented 18 US priorities to students, shown in table III. The priority were listed according to importance: 1-Low, 2-Medium and 3-High. Students had to understand that high priority US should be build first. Each US had specific requirements, demanding students to build them in a correct way and respecting their priority. If the US did not follow standards (e.g., colors, pieces, quantity or size), the PO could not accept it.

In Figure 5, the Kanban board with the PB is presented. As workshops were performed in different months, the US presentation was improved, by printing the Kanban in a large printed paper. In the first workshop (5a), the US was presented in the class board, in the last workshop (5c) it was presented in colored post-it.

After presenting the PB, each group performed a Planning Poker to estimate the required effort to build each US. In this step, DT analyzed each US and defined a value based in the Fibonnacci sequence, which 1 means low effort due to low complexity, followed by values 2, 3, 5 and 8. Effort 8 being

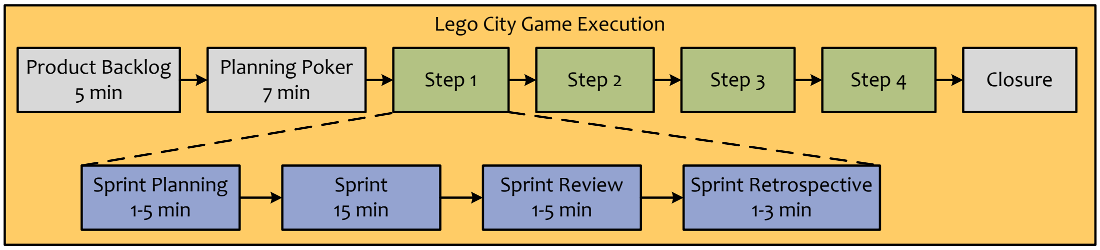


Fig. 4. Sequence of steps by Lego City Game execution based in Scrum methodology.

TABLE III
USER STORIES LIST OF PO'S PRODUCT BACKLOG WITH RESPECTIVE PRIORITIES.

| US | Priority | Item | Quantity | US | Priority | Item | Quantity |
|---|---|---|---|---|---|---|---|
| 1 | 3 | School | 1 | 10 | 1 | Small Warehouse | 1 |
| 2 | 3 | Church | 1 | 11 | 2 | Ambulance | 1 |
| 3 | 3 | City Hall | 1 | 12 | 2 | Fire Truck | 1 |
| 4 | 3 | Police Station | 1 | 13 | 2 | Police Car | 1 |
| 5 | 3 | Hospital | 1 | 14 | 1 | Bus | 1 |
| 6 | 3 | Movie Theater | 1 | 15 | 2 | School Bus | 1 |
| 7 | 2 | Super Market | 1 | 16 | 1 | Park Sculpture | 1 |
| 8 | 2 | Commercial Builder | 1 | 17 | 1 | Bus Station | 5 |
| 9 | 2 | Big Warehouse | 1 | 18 | 1 | Sports Court Accessories (School) | 3 |

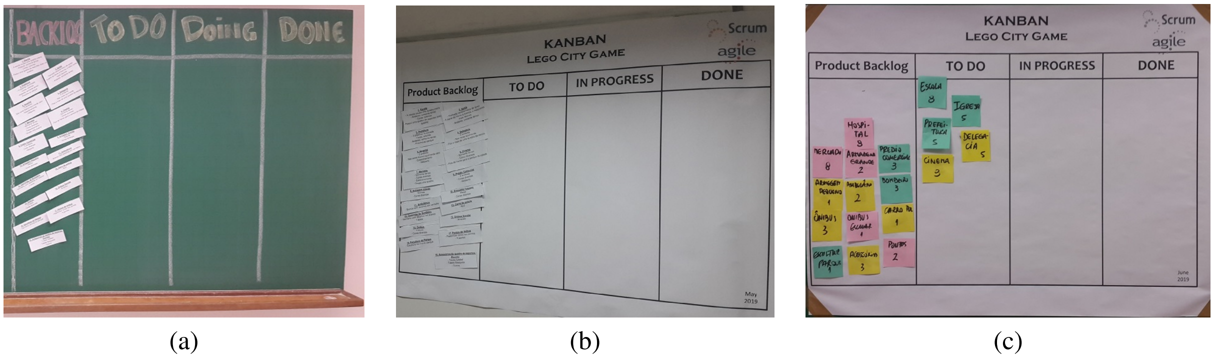


Fig. 5. Kanban board with US of PO by (**a**) first Workshop, (**b**) second Workshop and (**c**) third Workshop.

the highest one, due to high complexity. We consider that in each workshop the effort could be estimated in different values according to the DT perspective. So, an average was estimated (i.e, if a US had an estimated effort of 2, 2 and 3, we considered the effort average of 2). Based in estimative, we summed all US effort in order to have the total effort of the city building activity and used it in the Burndown chart, as showed in Figure 6. Although five sprints were available to students in each workshop, it was not necessary in any stance.

Before starting each Sprint, we performed a SP, in which the PO took the high priority US and requested that one group accepted the responsibility of building it. Each work group verified if they had the capability of building it. In this case one of the groups accepted each US requirement. In some cases, one group could accept US that required more effort than other teams, which did not represent a problem since all teams were able to complete the full tasks.

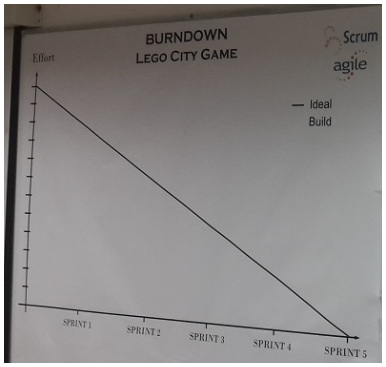


Fig. 6. Burndown chart used in Workshops.

During Sprints, each group started to build their respective US, with its progress being constantly updated in the Burndown chart from "To Do" to "In progress", and from "In progress" to "Done". In the final Sprint, the PO, DT ad SM met and performed the SRv in order to check each US from each group. If a task was performed correctly, the PO accepted it and declared the US finished. In case a US was rejected by the PO, the US was then rescheduled to a future Sprint. When a US was finished, the Effort value in the Burndown chart was decreased in order to show the Development Team performance. Before the next step, every team met in a SRt in order to analyze how to improve their performance. The next sprint then started and this process was repeated until Effort reached zero.

After all iterations were done, each student listed the main advantages of Scrum. In order to validate the learning process, we collected feedback using a form. The questions are presented in Table IV.

TABLE IV
EVALUATION QUESTIONNAIRE

| Question and Description | Alternatives |
|---|---|
| Q1 - Did you already know or heard about Scrum before participate this workshop? | Yes-Not |
| Q2 - Have you done any academic or professional Scrum practice before participate this workshop? | Yes-Not |
| Q3 - Based in a scale 1 to 5, how did you consider your knowledge about Scrum before the dynamics of Lego City Game? | Very low (1) - Very high (5) |
| Q4 - Based in a scale 1 to 5, how do you consider your knowledge about Scrum after the Lego City Game? | Very low (1) - Very high (5) |
| Q5 - Check out the roles of Scrum stakeholders you knew before performing the Lego City Game. | SM, PO, DT or Not know of them |
| Q6 - Check out the roles of Scrum stakeholders that you learned about who developed the Lego City Game. | SM, PO, DT or Not know of them |
| Q7 - Check out the Scrum artifacts you knew before performing the Lego City Game. | PB, SB, Kanban, Burndown, SRv, SRt or Not know of them |
| Q8 - Check out the Scrum artifacts you've come to know after performing the Lego City Game. | PB, SB, Kanban, Burndown, SRv, SRt or Not know of them |
| Q9 - Based in a scale 1 to 5, how do you consider the application of active activities like what was done with Scrum Lego City Game in teaching Agile methodologies? | Very Useless (1) - Very Useful (5) |
| Q10 - Based in a scale 1 to 5, do you agree that Lego City Game allowed for a greater understanding of Scrum when compared to just a theoretical class? | Totally disagree (1) - Totally agree (5) |
| Q11 - Based in a scale 1 to 5, do you consider that this game is a competitive differential in your academic background when compared to a student who only knew the Scrum methodology in theory? | Totally disagree (1) - Totally agree (5) |
| Q12 - Among the alternatives below, check out the options that you think might be a lot more effective in Scrum teaching and understanding when compared to Lego City Game. | Several Options |
| Q13 - From the alternatives below, mark the best option for a student who has never heard of Scrum teach and learn methodology more quickly. | Several Options |

Notes that Questions 3 and 4 measure the level of knowledge before and after the game. Both questions use the Likerd Scale to ask for students if acknowledgment is Very low (1), Low (2), Neither high nor low (3), High (5) or Very high (5) [40]. Also, at Questions 5, 6, 7 and 8 ask about what roles and artifacts students knowing before and after the game.

Question 12 includes options such as: book, article or manual (reading activities); video lessons; Audio lessons; Classroom theoretical teaching; real projects using a development time; real or online games; applications for smartphones; software emulation or educational simulation; group discussion; educational games; at work or internship; scrum certification events; seminar or lectures. Question 13 includes options such as: assisting a class, reading a technical article on scrum, achieving a practical academic or professional activity, or watching a video.

# V. RESULTS

In this section we present results divided into two parts. First, we describe the activities and results observed during the workshops in order to show how students interacted within the activity. Finally, the feedback collected are evaluated and discussed in order to present how the students acquired knowledge on Scrum and how the Lego city building activity improved learning.

## A. Workshop Activity

Initially the groups performed a Planning Poker meeting to define the required effort to perform eighteen US defined by the PO. Table V shows the estimated effort to for each DT in three workshops. The total Effort was applied on each Burndown chart. During Planning Poker, we observed that students were concerned with various aspects of defining Effort values. In particular, some students tried to estimate Effort based in the number of blocks that would be required to build each US. This was considered important to make decisions that would impact performance. The teamwork between stakeholders was also one of the main advances in this stage of Lego City. Some students were afraid to talk about the US effort, but after a few minutes they started interacting with other members. In all cases, a sense of leadership of some members improved task progress in all workshops.

TABLE V
PLANNING POKER WITH ESTIMATED EFFORT OF PRODUCT BACKLOG.

| US | DT 1 | DT 2 | DT 3 |
|---|---|---|---|
| 1 | 3 | 5 | 8 |
| 2 | 5 | 3 | 5 |
| 3 | 5 | 5 | 5 |
| 4 | 3 | 3 | 5 |
| 5 | 8 | 8 | 8 |
| 6 | 3 | 2 | 3 |
| 7 | 3 | 3 | 8 |
| 8 | 5 | 5 | 3 |
| 9 | 3 | 2 | 2 |
| 10 | 2 | 1 | 1 |
| 11 | 1 | 1 | 2 |
| 12 | 1 | 1 | 3 |
| 13 | 1 | 1 | 1 |
| 14 | 1 | 3 | 3 |
| 15 | 1 | 1 | 1 |
| 16 | 2 | 1 | 1 |
| 17 | 2 | 1 | 2 |
| 18 | 1 | 3 | 3 |
| **Total Effort** | **50** | **49** | **64** |

After defining the required Effort of all US, each team started using blocks to build the US. However, the choice of which US would be built in each Sprint considered the priority established by the PO and the team's ability to develop the US. Table VI shows results of what US were built in each Sprint. We can observe that on the first and on the second workshop, the DT accepted fewer US at Sprint 1 than Sprint

2, this is because teams had no real knowledge of their building capacity. In this sense, Sprint 2 was more critical because it showed the outcome of performance assumptions made in Sprint 1. The first Sprint of the third workshop presented more US effort than other stances. We can relate this high Effort value due to the number of students being 27 (in comparison to 14 and 20). Due to the high number of students, the US were build more quickly, needing only three Sprints to conclude the project.

TABLE VI
STUDENTS PERFORMANCE IN EACH WORKSHOP.

| Sprint | Total Effort DT 1 | Total Effort DT 2 | Total Effort DT 3 |
|---|---|---|---|
| 1 | 13 | 16 | 26 |
| 2 | 22 | 18 | 18 |
| 3 | 9 | 6 | 20 |
| 4 | 6 | 9 | - |
| **Total Effort** | **50** | **49** | **64** |

Figure 7 shows the Burndown charts of each DT. In all cases, students got to build all US and were motivated to use Scrum concepts in order to create the Lego blocks city. One of the teaching challenges was to make students understand the importance of updating the Kanban. It was necessary to warn students not to forget to update the status (To do, In progress, Done) of each US. Students became more confident as US were update in real time, allowing visualization of each activity.

Figure 8 shows the final Lego City. One of the great advantages of these workshops is not exactly the fact students are learning new contents or a new methodology, but rather the way activities are accomplished. A creative and entertaining environment makes students feel outside an academic mental state, improving their learning process due to unusual techniques. Creative environments are positively accepted, especially among people aged 18 to 25, because they are used to fast changing concepts like technology and electronic games. A Scrum-based game seeks to create an interactive and collaborative environment among students. Among other developed trait, we can mention: responsibility, team work, good design, adherence to deadlines, workspace organization, respect to the methodology stages and face-to-face communication.

### B. Students Feedback

Initially, any student could participate in the workshops, without any previously knowledge of Scrum being mandatory. In this context, it was verified that 93.6% of all students had some knowledge about scrum. When asked if they had previously participated in a Scrum activity, we verified that 66% of students had some experience in Scrum workshops or tasks.

Based on the first and second questions, we asked students to consider their level of Scrum knowledge before and after participating in the workshop. Table VII presents the results obtained from student's feedback. It is possible to verify that before the workshop, students who considered a knowledge level of 1, 2 or 3 was 74.5% and after the workshop the percentage was reduced to 8.5%. These feedbacks show how much the workshop increased their knowledge. Students who considered their knowledge to be high (4) or very high (5) went from 25.5% to 91.5% after the workshop. There were no cases of students who reduced their level of knowledge, with most of them going up 1 or 2 levels.

TABLE VII
SCRUM LEVEL OF KNOWLEDGE BEFORE AND AFTER WORKSHOPS.

| Scrum Level of Knowledge | Before | After |
|---|---|---|
| Very low (1) | 12.8% | 0.0% |
| Low (2) | 25.5% | 2.1% |
| Normal (3) | 36.2% | 6.4% |
| High (4) | 21.3% | 40.4% |
| Very high (5) | 4.2% | 51.1% |

Table VIII shows the results of questions 5 and 6. Students were asked about which Scrum stakeholder they had known before and which ones they came to know after the workshop. Among students who had no knowledge of Scrum, the results were positive. Unfortunately, many students indicated they knew "Development Time" because they had some experience in academic projects that used this term and related that to Scrum specific concept of Development Time. Due to the possibility of selecting more than one option, the sum of values can be greater than 100%.

TABLE VIII
PERCENTAGE OF SCRUM STAKEHOLDER THAT STUDENT KNOW BEFORE AND AFTER WORKSHOPS.

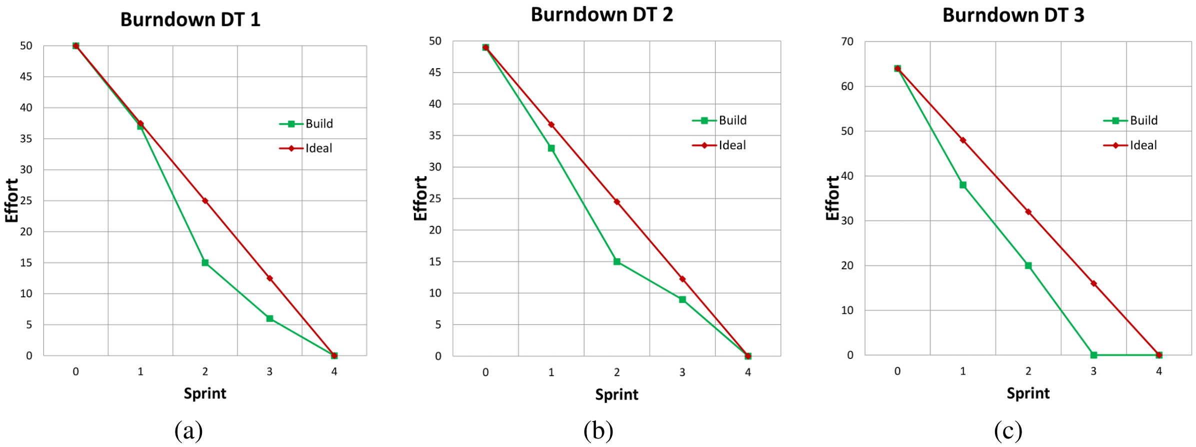
| Scrum Stakeholder | Before | After |
|---|---|---|
| Scrum Master | 85.1% | 93.6% |
| Product Owner | 72.3% | 93.6% |
| Development Team | 76.6% | 97.9% |
| Not know of them | 4.3% | 2.1% |

The variety of elements in Scrum can represent a challenge for students. In questions 7 and 8, we evaluated which Scrum tools the students knew before participating. Students were able to choose more than one answer, resulting in a sum greater than 100%. In table IX, the results of these questions can be observed. In all cases, students indicated an increase of knowledge on Scrum tools. Some students who did not know scrum indicated they knew Kanban due to the use of this technique in other scenarios, e.g., companies, factories and offices. The fastest growing elements were Burndown (from 34% to 85.1%), Sprint Retrospective (from 40.4% to 80.9%) and Sprint Review (from 46.8% to 87.2%). This is due to the fact that these elements are rarely presented to students during theoretical and practical activities.

Based on the results in Tables VII, VIII and IX it is possible to affirm that students' feedback indicates an improved understanding of Scrum topics after the game.

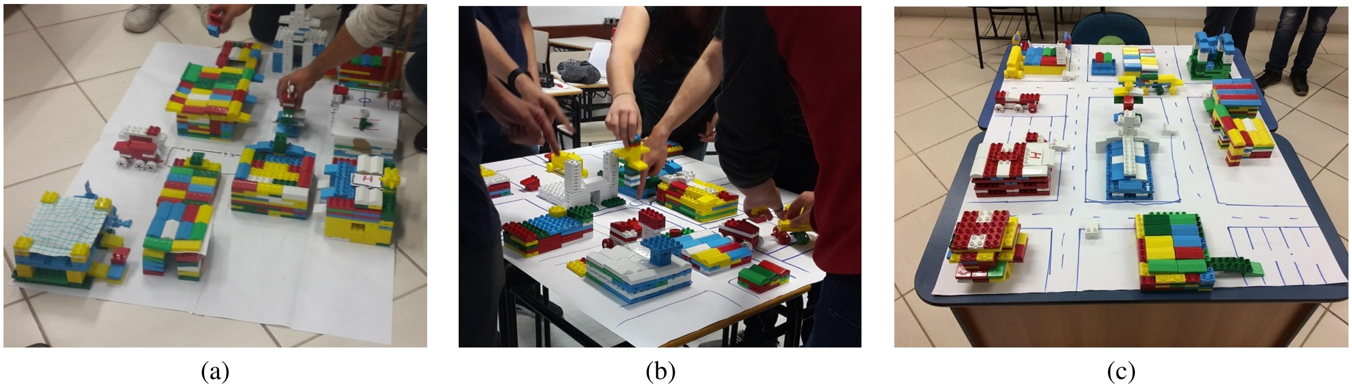
Figure 9 shows results obtained from questions Q9, Q10 and Q11. Scales are presented in values from 1 to 5, in which 1 represents useless and 5 represents very useful, following

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/RITA.2020.2987721, IEEE Revista Iberoamericana de Technologias del Aprendizaje



(a) (b) (c)

Fig. 7. Burndown charts performance (**a**) first Workshop, (**b**), second Workshop and (**c**) third Workshop.



(a) (b) (c)

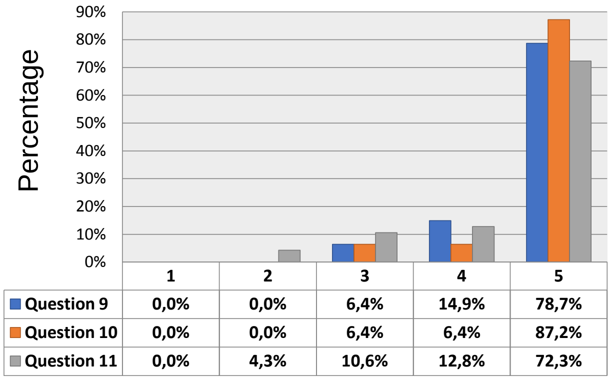
Fig. 8. Lego City of Students: (**a**) first Workshop, (**b**) second Workshop, and (**c**) third Workshop.

TABLE IX
PERCENTAGE OF SCRUM ARTIFACT THAT STUDENT KNOW BEFORE AND AFTER WORKSHOPS.

| Scrum Artifact | Before | After |
|---|---|---|
| Product Backlog | 72.3% | 87.2% |
| Sprint Backlog | 72.3% | 87.2% |
| Kanban | 78.7% | 80.9% |
| Burndown | 34.0% | 85.1% |
| Sprint Retrospective | 40.4% | 80.9% |
| Sprint Review | 46.8% | 87.2% |
| Not know of them | 8.5% | 0.0% |

question 9. In questions 10 and 11, values represent 1 (fully disagree) up to 5 (fully agree). Question 9 shows that all students consider using activities like Lego City to teach Scrum a very effective method. We confirm this feedback in Q10 that shows more that more than 85% of students feel that learning Scrum sing Lego City is more effective than lectures, slides and theoretical classes. When asked in question 11 if students consider Lego City to be competitive advantage in comparison to students who only had theoretical classes, the results have varied. Students who fully agreed were majority with about 72.3%, but students also indicated replies number 2, 3, and 4, with 4.3%, 10.6% and 12.8% respectively.

Prospects of activities that may aid student learning are presented in Table X. Question Q12 ask students about the different ways the Scrum learning process can be improved, by presenting different alternatives. Results showed that about 70% of all students chose real project activities with a real development team, and 66% chose at the work or during an internship. The activities considered less effective were: the reading of books and articles and audio-classes. Furthermore, we observed that only 12.8% of opinions are related to theory classes, showing that traditional teaching might generate low motivation and effectiveness. These results are mainly based on students' feeling. It is worth mention that it was not proposed in this work to apply other tools to teach Scrum



| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 🟦 Question 9 | 0,0% | 0,0% | 6,4% | 14,9% | 78,7% |
| 🟧 Question 10 | 0,0% | 0,0% | 6,4% | 6,4% | 87,2% |
| ⬜ Question 11 | 0,0% | 4,3% | 10,6% | 12,8% | 72,3% |

Fig. 9. Feedback of questions Q9, Q10 and Q11.

such as watching videos, developing real projects under Scrum process, reading books, having tutorial classes, among others.

TABLE X
STUDENTS PERSPECTIVES ABOUT BETTER WAYS TO LEARNING SCRUM METHODOLOGIES.

| Method of learning Scrum | Percentage |
|---|---|
| Real project in a development time | 70.2% |
| At work or internship | 66.0% |
| Game or online game | 40.4% |
| Present educational games | 27.7% |
| Application for smartphone | 23.4% |
| Video lessons | 19.1% |
| Seminar or lecture | 17.0% |
| Carrying out scrum certification tests | 12.8% |
| Software, emulator or educational simulator | 12.8% |
| Present discussion groups | 12.8% |
| Theory in a classroom by a professor | 12.8% |
| Book, article or manual (read only) | 6.4% |
| Audio classroom | 6.4% |

Finally, in question 13, students were asked about the best alternative to learn Scrum quickly. Students could choose only one alternative. Results show that 91.5% of students consider practical activities in academic or professional environments are more effective than lectures (4.3%), reading books (0%) or watching videos (4.3%).

### C. Comparative Works

Accordingly, the evaluation objective is to analyze the game with Lego blocks with respect to motivation for learning in the context of computer courses. Based on the perspective of students' motivation observed from the questionnaires, we performed a comparative study with the previous works indicated in section III. A comparison of results based on the motivation of students is shown in Table XI, in which we focused at Questions 9, 10 and 11 where students could evaluate how the game improved the learning and teaching of activity.

At works [14], [17] and [35] we focus only on where students checked a high level of agree (5) with the question (Q9, Q10, and Q11) considering the lowest level (1) up to (5). Based on the responses of students it is possible to see that our proposed work presents better or equal results than [14] and [17]. At work [35] the evaluation of students present a better motivation for Q9 and Q11, but the number of students is lower than our proposed work.

The work [15] did not present any numerical results, but show opinions about the motivation and how the students feel about the application of activity. In [15] and in the proposed work, the students referred to the activity as fun, challenging, well applied, very well organized and easy to understand the Scrum concepts. In the original paper, other students' feedbacks are available.

Participant students from the work [16] indicated their motivation using the Likerd scale [40] score notes (1 - worst up to 5 better). In our questionnaire, we adapt these scales to totally disagree (1) up to totally agree (5). Our results present better values when compared to [16], as shown in Table XI.

## VI. CONCLUSIONS

In this article we presented the results of applying an academic activity based in the creating of a Lego blocks city. The construction of the city uses steps, stakeholders and elements of the agile Scrum methodology. The experiment was done in three different workshops, with the participation of students. Initially, students were guided in the different stages of the Scrum and then were divided into groups for the construction of eighteen activities of the PB elaborated by the PO. In the end the students answered a questionnaire to understand what their perception was in the development of the workshop.

Results show that students considered the activity very effective to learn Scrum. When analyzing other techniques, students also indicated that the practicality of activities such as the Lego blocks city is a good alternative.

As further works we encourage the analysis of learning activities, including Lego blocks, for methodologies such as XP, FDD, Lean or Kanban Software developments. Most of the related works presented the motivation feelings and encouragement of students at proposed activities, however, it does not provide any comparative with other traditional learning processes, needing to study with complex research of many activities compared to Lego City Game. Still, it is suggested to correlate the work interaction between professors and students during the learning process, including the feeling motivation of professors to use this activity more than other traditional learning activities. The leading role of students can be analyzed in this activity, i.e., Scrum master or leader team student, in order to manage the team, perform deadlines, fix construction problems, mitigate lower motivation of the team, or propose solutions for challenges. In order to evaluate specific topics of Scrum it is suggested to apply tests to students using Scrum Certification exams, and not only evaluate the students' intuition perceptions.

## REFERENCES

[1] L. Osterweil, "Software processes are software too," in *Engin. of Soft.* Springer, 2011, pp. 323–344.

[2] Y. B. Leau, W. K. Loo, W. Y. Tham, and S. F. Tan, "Software development life cycle AGILE vs traditional approaches," in *Inter. Conf. on Inf. and Netwo. Techn.*, vol. 37, no. 1, 2012, pp. 162–167.

[3] R. Fojtik, "Extreme Programming in development of specific software," *Proc. Comp. Scie.*, vol. 3, pp. 1464–1468, 2011.

[4] B. Boehm and R. Turner, "Management challenges to implementing agile processes in traditional development organizations," *IEEE Soft.*, vol. 22, no. 5, pp. 30–39, 2005.

[5] S. Nerur, R. Mahapatra, and G. Mangalaraj, "Challenges of migrating to agile methodologies," *Commun. of the ACM*, vol. 48, no. 5, pp. 72–78, 2005.

[6] C. Scharff, S. Heng, and V. Kulkarni, "On the difficulties for students to adhere to Scrum on Global Software Development projects: Preliminary results," in *2nd Intern. Works. on Collab. Teach. of Glob. Dist. Softw. Devel.*, June 2012, pp. 25–29.

[7] V. Vinekar, C. W. Slinkman, and S. Nerur, "Can agile and traditional systems development approaches coexist? An ambidextrous view," *Infor. Syst. Manag.*, vol. 23, no. 3, pp. 31–42, 2006.

[8] V. Devedzic and S. R. Milenkovic, "Teaching Agile Software Development: A Case Study," *IEEE Trans. on Educ.*, vol. 54, no. 2, pp. 273–278, May 2011.

[9] V. M. F. Fonseca and J. Gmez, "Applying Active Methodologies for Teaching Software Engineering in Computer Engineering," *IEEE Rev. Ibero. de Tecn. del Aprend.*, vol. 12, no. 3, pp. 147–155, August 2017.

TABLE XI
COMPARATIVE RESULTS WITH RELATED WORKS.

| Related work | Number of students | Results of related works | Results of our proposed work |
|---|---|---|---|
| [14] | 73 | Answer to strongly agree (5) by question<br>Q9 - 32% / Q10 - 40% / Q11 - 35% | Answer to totally agree (5) by question<br>Q9 - 78.7% / Q10 - 87.2% / Q11 - 73.3% |
| [15] | - | "*I liked how something so fun and challenging ended with a lesson that I will never forget... initial estimations are always wrong... so do your best but don't break your back!*" | *"I found the dynamics fun, well applied and organized. For me it was much easier to understand the concepts of Scrum"*. Student opinion. |
| [16] | 15 | Score note from 1 (totally disagree) to 5 (totally agree)<br>Q9 - 4.4 / Q10 - 4.5 / Q11 - 4.3 | Score note from 1 (totally disagree) to 5 (totally agree)<br>Q9 - 4.7 / Q10 - 4.8 / Q11 - 4.5 |
| [17] | 45 | Answer to strongly agree (5) by question<br>Q9 - 78% / Q10 - 28.9% | Answer to totally agree (5) by question<br>Q9 - 78% / Q10 - 87.2% |
| [35] | 26 | Answer to strongly agree (5) by question<br>Q9 - 82% / Q10 - 80% / Q11 - 80% | Answer to totally agree (5) by question<br>Q9 - 78.7% / Q10 - 87.2% / Q11 - 73.3% |

[10] J. L. Duffany, "Application of Active Learning Techniques to the Teaching of Introductory Programming," *IEEE Rev. Ibero. de Tecn. del Aprend.*, vol. 12, no. 1, pp. 62–69, February 2017.

[11] R. M. Felder and L. K. Silverman, "Learning and teaching styles in engineering education," *Engin. Educ.*, vol. 78, no. 7, pp. 674–681, 1988.

[12] M. J. Prince and R. M. Felder, "Inductive teaching and learning methods: Definitions, comparisons, and research bases," *Jour. of Engin. Educ.*, vol. 95, no. 2, pp. 123–138, 2006.

[13] S. J. Armstrong and C. V. Fukami, *The SAGE handbook of management learning, education and development*. Sage, 2009.

[14] C. G. Von Wangenheim, R. Savi, and A. F. Borgatto, "SCRUMIA - An educational game for teaching SCRUM in computing courses," *Jour. of Syst. and Soft.*, vol. 86, no. 10, pp. 2675–2687, 2013.

[15] J. May, J. York, and D. Lending, "Teaching tip: Play ball: Bringing scrum into the classroom," *Jour. of Inform. Syst. Educ.*, vol. 27, no. 2, p. 87, 2016.

[16] W. L. LEE, "SCRUM-X: An interactive and experiential learning platform for teaching scrum," in *Proc. of the 7th Intern. Confer. on Educ., Train. and Inform.* Elsevier, 2016.

[17] G. Rodriguez, Á. Soria, and M. Campo, "Virtual Scrum: A teaching aid to introduce undergraduate software engineering students to Scrum," *Comp. Applic. in Engin. Educ.*, vol. 23, no. 1, pp. 147–156, 2015.

[18] A. Godoy and E. F. Barbosa, "Game-Scrum: An approach to agile game development," in *Proc. of SBGames*, 2010, pp. 292–295.

[19] C. Scharff and R. Verma, "Scrum to support mobile application development projects in a just-in-time learning context," in *Proc. of the Works. on Cooper. and Human Aspec. of Soft. Engin.* ACM, 2010, pp. 25–31.

[20] C. A. G. da Silva, E. L. dos Santos, L. M. Angelo, R. V. de Moraes *et al.*, "A utilização do SCRUM como recurso educacional no processo de aprendizagem em Engenharia de Software," *Intern. Jour. on Alive Engin. Educ.*, vol. 3, no. 2, pp. 87–102, 2016.

[21] H. Takeuchi and I. Nonaka, "The new new product development game," *Harvard business review*, vol. 64, no. 1, pp. 137–146, 1986.

[22] C. Larman and V. R. Basili, "Iterative and incremental developments. a brief history," *Computer*, vol. 36, no. 6, pp. 47–56, 2003.

[23] K. Schwaber, "Scrum development process," in *Business object design and implementation*. Springer, 1997, pp. 117–134.

[24] J. Highsmith and A. Cockburn, "Agile software development: The business of innovation," *Computer*, vol. 34, no. 9, pp. 120–127, 2001.

[25] V. Mahnič and T. Hovelja, "On using planning poker for estimating user stories," *Jour. of Syst. and Soft.*, vol. 85, no. 9, pp. 2086–2095, 2012.

[26] R. Tamrakar and M. Jørgensen, "Does the use of Fibonacci numbers in planning poker affect effort estimates?" in *16th Intern. Conf. on Eval. Asses. in Soft. Engin.*, May 2012, pp. 228–232.

[27] V. Mahnic, "A capstone course on agile software development using Scrum," *IEEE Trans. on Educ.*, vol. 55, no. 1, pp. 99–106, 2011.

[28] M. Paasivaara, J. Vanhanen, V. T. Heikkilä, C. Lassenius, J. Itkonen, and E. Laukkanen, "Do high and low performing student teams use scrum differently in capstone projects?" in *39th Intern. Conf. on Soft. Engin.: Soft. Engin. Educ. and Train. Track*. IEEE, 2017, pp. 146–149.

[29] M. Golfarelli, S. Rizzi, and E. Turricchia, "Sprint planning optimization in agile data warehouse design," in *Inter. Conf. on Data Wareh. and Knowl. Discov.* Springer, 2012, pp. 30–41.

[30] Z. Azham, I. Ghani, and N. Ithnin, "Security backlog in Scrum security practices," in *Malays. Conf. in Soft. Engin.* IEEE, 2011, pp. 414–417.

[31] L. Rising and N. S. Janoff, "The Scrum software development process for small teams," *IEEE Software*, vol. 17, no. 4, pp. 26–32, 2000.

[32] C. A. G. da Silva, E. L. dos Santos, and D. A. F. Pelacini, "Evaluation of Academic Experience in Learning Education over Simulators Softwares," *Inter. Jour. on Alive Engin. Educ.*, vol. 5, no. 2, pp. 23–40, 2018.

[33] M. Kropp, A. Meier, M. Mateescu, and C. Zahn, "Teaching and learning agile collaboration," in *IEEE 27th Confer. on Soft. Engin. Educ. and Train.* IEEE, 2014, pp. 139–148.

[34] J.-P. Steghöfer, H. Burden, H. Alahyari, and D. Haneberg, "No silver brick: Opportunities and limitations of teaching scrum with lego workshops," *Jour. of Syst. and Soft.*, vol. 131, pp. 230–247, 2017.

[35] R. Castro *et al.*, "AGILITY SCRUM–Um Jogo para Ensino da Metodologia SCRUM," in *25$^{o}$ WEI-Workshop sobre Educação em Computação*, 2017.

[36] J. Marty and T. Carron, "Observation of Collaborative Activities in a Game-Based Learning Platform," *IEEE Trans. on Learn. Techn.*, vol. 4, no. 1, pp. 98–110, Jan 2011.

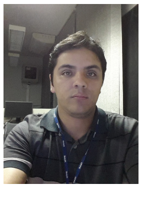
[37] K. Getchell, A. Miller, J. R. Nicoll, R. Sweetman, and C. Allison, "Games Methodologies and Immersive Environments for Virtual Fieldwork," *IEEE Trans. on Learn. Techn.*, vol. 3, no. 4, pp. 281–293, October 2010.

[38] . Serrano-Laguna, J. Torrente, B. M. Iglesias, and B. Fernndez-Manjn, "Building a Scalable Game Engine to Teach Computer Science Languages," *IEEE Rev. Ibero. de Tecn. del Aprend.*, vol. 10, no. 4, pp. 253–261, November 2015.

[39] M. Paasivaara, V. Heikkilä, C. Lassenius, and T. Toivola, "Teaching students scrum using LEGO blocks," in *Proc. of the 36th Intern. Confer. on Soft. Engin.* ACM, 2014, pp. 382–391.

[40] R. Likert, "A technique for the measurement of attitudes," *Arch. of Psych.*, 1932.

**Douglas Augusto Barcelos Bica** received his Bachelor degree in Chemistry from the Federal University of Paraná (UFPR) in 2016. Currently he is a Undergraduate degree student of Analysis and Development of System technological course at University of Araucária (Unifacear) in Brazil. His main areas of interest are software development, agile methodologies, Scrum Framework, and Computer Education.



**Carlos Alexandre Gouvea da Silva** (SM'16-M'18) received his B.Eng. degree in Computer Engineering from the Pontifical Catholic University of Paraná in 2012 and his Master degree in Electrical Engineering in 2015 in Federal University of Paraná, where he is currently pursuing the Ph.D. degree. Research interests include computer networks, video transmission and coding, wireless networks, education and quality of teaching in engineering.