

Learning Software Quality Assurance with Bricks

Miguel Ehécatl Morales-Trujillo
University of Canterbury
Christchurch, New Zealand
miguel.morales@canterbury.ac.nz

Abstract—Software Quality Assurance (SQA) and Software Process Improvement (SPI) are topics of crucial importance for software engineers; however, teaching them in a lecture room comes with several limitations due to lack of practical experience. With that in mind, we created KUALI-Brick, a LEGO®-based activity that brings SQA and SPI concepts together applying them in order to successfully build a LEGO city. This hands-on activity has been carried out in a fourth-year Software Engineering course at the University of Canterbury, with current results showing high levels of fun, increased engagement and an improved learning experience. We present a step-by-step guide to replicate the activity as well as lessons learned after conducting the activity for three consecutive years.

I. INTRODUCTION

Over the years, the term quality has been defined from various perspectives. For example, it has been seen as the “conformance to requirements” [1] for “achieving excellent levels of fitness for use” [2] where “the customer is the final arbiter and it is market-driven” [3].

Nonetheless, in the words of Pfleeger and Kitchenham, “quality is hard to define, impossible to measure, easy to recognise” [4]. According to Gillies, “quality is generally transparent when present, but easily recognized in its absence” [5]. Based on these definitions, it is evident that quality, although purely abstract, must be instantiated in some way in order to be demonstrated.

In the context of Software Engineering, the term “software quality” is defined as “the “capability of a software product to satisfy stated and implied needs under specified conditions” [6]. Software quality is “the degree to which a system, component, or process meets (i) specified requirements, and (ii) customer or user needs or expectations” [7].

The concept of software quality is crucial for software engineers, who have to ensure that software development processes they follow together with software products they create are of high quality. In other words, software engineers have no choice but to show that their processes and products are suitable and conform to the specified requirements and expectations. In order to prove this suitability and compliance, Software Quality Assurance (SQA) and Software Process Improvement (SPI) play a fundamental role in software quality.

While SQA is defined as “a planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements” [8], SPI represents “a planned, managed and controlled effort which aims to enhance the capability of the software development processes of an organization” [9].

Both, SQA and SPI, cannot be taught by applying a single standard approach or a single set of common procedures as students need hands-on experience with practical scenarios [10]. Following the motivation of providing Software Engineering students with hands-on practice in the classroom, in this experience report we present KUALI¹-Brick: a LEGO®-based activity that aims at understanding and applying SQA and SPI concepts and techniques while participants build objects and assess their quality based on metrics associated to specific quality attributes.

On the one hand, we have chosen LEGO as it offers the means to make quality concepts tangible. On the other hand, activities that involve playing in the classroom offer several important benefits in order to enhance the learning process [10].

We also present some preliminary empirical results and lessons learned after having successfully performed KUALI-Brick activity for the last three years with fourth-year Software Engineering students in a SQA course at the University of Canterbury, New Zealand.

The remainder of this paper is structured as follows: Section II describes current work on the use of LEGO in Software Engineering education. Section III and IV expand on the SQA course description and the KUALI-Brick activity respectively. Section V consolidates discussion of the results. Finally, conclusions and future work are presented in Section VI.

II. RELATED WORK

In this section we present results of a literature review on the topic as well as a summary of the papers found.

A. Literature Review

While LEGO popularity is indisputable in the toy industry, its usage has definitely crossed boundaries to other fields with the education sector being one of the most striking examples. In the context of this paper, our interest is focused on identifying those educational activities that use LEGO in Software Engineering. This motivated the following research question:

How is LEGO used in Software Engineering education?

To find answers to the aforementioned question, we performed a literature review with a search run in four databases: Scopus, IEEE Xplore, ACM Digital Library and Wiley Online Library. The search string was the following:

¹KUALI: Nahuatl word meaning good, fine or appropriate.

“lego AND education”

The search scope was focused on peer-reviewed research papers published in journals, conferences and workshops. The inclusion criteria applied for the selection procedure were:

- Papers that satisfy the search string.
- Papers published in journals, conferences or workshops.
- Papers written in English.

The exclusion criteria considered were:

- Papers with no focus on Software Engineering education.
- Papers dealing with programming or robotics.

The last exclusion criteria turned out to be necessary after obtaining a considerably high number of papers that address LEGO Mindstorms in the context of programming and robotics. While these topics are related to Software Engineering, they are not considered to be within the scope of this research.

The search was executed on October 1st, 2020. The total number of returned papers was 776, with 241 from ACM Digital Library, 435 from Scopus, 92 from IEEE Xplore and 8 from Wiley Online Library. After applying the inclusion and exclusion criteria, 16 papers were identified as relevant (see Table I).

For the purposes of an accurate data extraction, a template containing fields for paper ID, authors, title, year of publication, document type, abstract, topic, data collection method applied, number of participants, and results was created. The findings of the literature review are presented in the following subsection.

B. Software Engineering Education with LEGO

Teaching Agile frameworks, particularly Scrum and its related concepts, is the most recurrent topic in which LEGO is being used. Lynch et al. created a boot camp for learning Agile software engineering concepts with LEGO bricks as the medium [11]. Authors reported that most of the students enjoyed the boot camp, however, their apparent retention of the information was not significantly better than any of the other course information [11].

Bica and da Silva [12] designed a classroom activity to teach Scrum concepts to undergraduate students. The authors reported that the knowledge of the students in regards to Scrum increased, same as their motivation.

Steghöfer et al. reported teaching Scrum with LEGO as a means of providing practical experience to anchor theoretical knowledge [13]. Their results showed that most students were positive about the activity and felt pleased with the hands-on experience.

Paasivaara et al. created a LEGO-based simulation game to support the students' learning of roles, events and concepts in Scrum [14]. The simulation game was evaluated with a survey and student lecture diaries. The authors concluded that the simulation game was a valuable tool for teaching Scrum to university students, the students' reaction to the game was positive, and the learning outcome improved [14].

LEGO bricks have also been used to introduce students and IT professionals into Agile and XP concepts [15], the main ideas and practices of TDD [16] and the use of Essence [17], a standard that aims at defining and consequently improving Software Engineering practices.

In [18], LEGO is used to introduce the concepts of SPI and Scrum into a Bachelor level course with the aim of providing students with a direct experience of observing issues, creating an improvement plan to address them, and finally, applying and evaluating the plan. Anecdotally, the KUALI-Brick activity presented in this paper originated when the author of the paper was teaching Kanban and Scrum to IT professionals in 2013, having evolved over the years.

In addition, LEGO Serious Play™ (LSP) has been used to teach several core software engineering topics through hands-on scenarios, such as requirements engineering [10] [19] [20] and risk identification [19]. Initial results suggest that LSP has a positive impact on student learning, while also increasing student engagement with the course material [19]. However, no significant improvement has been identified [10].

Gama [21] reported an active learning approach where LEGO bricks are used to exemplify mini-project scenarios with the objective to teach requirements engineering and project planning.

Teaching particular strategies or learning specific skills are other goals that have been pursued by means of LEGO. For example, a bridge building project has been reported in [22]; the project teaches students how to measure progress and manage change. Other examples are: allowing students to practice collaborative design, parallel development, and component integration [23].

Šablís et al. used LEGO bricks to help students understand the communication and coordination challenges, and problems that might arise in global software engineering (GSE) projects, with positive results [24].

Lastly, Laird and Yang presented a method that uses LEGO to allow students to get hands-on practice in their estimation skills and techniques as well as in testing some of the commonly held estimation heuristics [25].

To sum up, the primary papers reported studies involving from 15 to 200 participants. While surveys are the most common means of validation, tests, questionnaires, exercises and diaries have been used as data collection methods as well. A summary of the topics, number of participants and data collection methods is presented in Table I.

In addition to the literature review, a Google search on the topic was performed and one meaningful resource was found outside the scientific literature: LEGO for Software Engineering [26]. However, the activities published on this website are presented in the papers mentioned and discussed above [10] [16] [19] [23].

Overall, despite an evidently positive application of LEGO in delivering Software Engineering concepts and practices to university students and IT professionals, there is a clear need to provide sufficient detail on the methods used as well as

TABLE I
SUMMARY OF THE PRIMARY PAPERS

ID	Topic	Participants	Data collection method
[11]	Scrum	43	Survey
[12]	Scrum	15	Survey
[13]	Scrum	Not reported	Survey
[14]	Scrum	46	Survey and lecture diary
[15]	XP	41	Questionnaire
[16]	TDD	Not reported	Not reported
[17]	Software Processes	Not reported	Not reported
[18]	SPI	20	Survey
[19]	Requirements and Risk identification	20	Test
[20]	Requirements	~200	Survey
[10]	Requirements	21	Formal test
[21]	Requirements Project Planning	33 25	Survey Survey
[22]	Change Management	Not reported	Not reported
[23]	Construction	Not reported	Not reported
[24]	GSE	104	Survey
[25]	Estimation techniques	17	In-class exercise

to produce empirical evidence that would support the claims presented by the researchers.

III. THE COURSE: SENG403 – SOFTWARE PROCESS AND PRODUCT QUALITY

The SENG403 – Software Process and Product Quality course introduces software quality key concepts, practices, methodologies and techniques present through the software lifecycle. SENG403 is a fourth-year elective course for Software Engineering and Computer Science (Hons.) students. It is also available as an elective for Master students in Information Systems and Information Technology.

SENG403 has been taught in the Software Engineering programme at the University of Canterbury since 2018. The course has been designed by the author of the paper based on his experience in software industry and tertiary education since 2011. The topics covered in the course include the following:

- 1) **Quality basics.** An introduction to the basic concepts of quality, its relation to Software Engineering, the current obstacles and solutions for ensuring quality of software systems.
- 2) **Quality models and standards.** A review of quality standards, conventions and models used in the industry.
- 3) **Managing quality in software projects.** An overview of the quality management knowledge and techniques, including planning and control activities.
- 4) **Quality in software processes.** An in-depth review of the challenges faced by organizations developing software products, and how those are managed through SPI initiatives.

- 5) **Product quality.** A description of the quality approach focused on the quality attributes of a software product, the models, standards and validation and verification practices.
- 6) **Introduction to measurement.** An overview to the need of measure, its basic concepts, the standards and methodologies related with metrics and measurement.
- 7) **Tools and techniques.** A review of quality tools and techniques useful during management and assurance processes.

At the end of SENG403, students are expected to understand, design and conduct, from a broader perspective, actions that improve the quality attributes of what/how they develop. SENG403 has a total of nine learning outcomes (LOs), relevant for this paper being the following:

- LO1 – Demonstrate advanced knowledge of quality concepts. Describe, discuss, and apply the principles of SQA.
- LO2 – Apply analysis skills to abstract and devise quality problems that affect process, product and people (the P's) in the software engineering context.
- LO4 – Evaluate complex and integral systems in order to recognize failures.
- LO6 – Compare and evaluate available solutions and apply the most suitable strategy to improve quality aspects of the P's.

The full information of the course can be consulted at [27].

IV. KUALI-BRICK ACTIVITY

As part of SENG403, we carry out the KUALI-Brick activity towards the end of the course. The activity itself is described in detail in the following subsections.

A. Objective and Target Audience

The learning goal of the activity is to understand and apply the SQA and SPI concepts. In order to achieve this, students need to: (i) understand, adopt and adapt a predefined building process; (ii) analyze and define the most relevant quality attributes to be assessed; and (iii) discuss and agree on the metrics to be collected.

The purpose of the activity is to build a LEGO city within a fixed amount of time and obtaining as many points as possible. Points are allocated according to the quality assessment of the objects that the teams build.

The target audience of the activity consists of undergraduate Software Engineering students enrolled in the fourth year of their Bachelor degree.

B. Materials and Space

The participants have access to a wide variety of LEGO bricks, a predefined building process and a description of the expected objects to be built. The required material is the following:

- 12 boxes of LEGO bricks (approx 3,000 bricks in total), the bricks are organized by colour and size. See Figure 1.



Fig. 1. Boxes of bricks organized by colour and size.

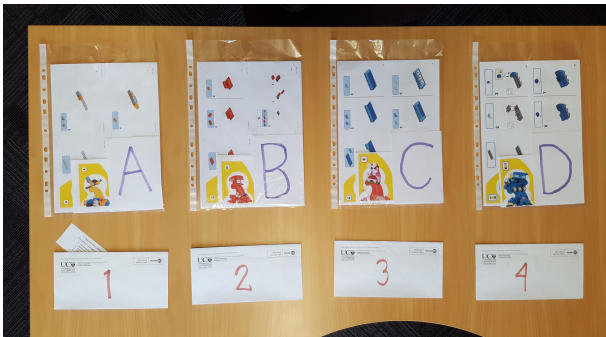


Fig. 2. Envelopes and pockets containing the building processes and the objects.

- 8 numbered envelopes containing building processes that each team will follow. There are three different building process “approaches”.
- 8 copy-safe pockets, each containing a set of four LEGO objects that each team will have to build. The pockets are labeled with letters. See Figure 2.
- A city plan (optional).
- A card with the steps of the activity for each team (optional).
- A whiteboard, markers and a stopwatch.
- A projector (optional).

This material is sufficient for 32 students distributed in eight 4-member teams; however, the ideal number is 24 students distributed in six teams.

The space required for the activity is a flat room, ideally with a projector. A table per team is required with two bigger tables to sit the boxes with the bricks and the city plan, placed in the way to be easily accessible by the participants. The participants are distributed in teams around the room, while the bricks and the city plan are located in the middle.

C. Activity Step-by-Step

The KUALI-Brick session is designed to be carried out in 100 minutes and consists of three to four iterations. Three main stages can be identified: the Set-up and presentation stage is followed by the Building-and-Assessment iterations, with a wrap-up during the Discussion stage.

During the Set-up and presentation stage, the instructor sets up the materials and arranges the room, presents the

generalities of the activity, explains the rules and divides the students into teams. The steps for this stage are listed below with suggested times for each step included in brackets:

- 0) The instructor presents the learning goal and objective of the activity.
- 1) The participants are asked to form teams of at most four people.
- 2) As a team, they choose one of the copy-safe pockets labeled from A to H. It contains a set of LEGO objects that they will have to build.
- 3) As a team, they discuss common quality attributes and their associated metrics that a LEGO build should have. Three to five attributes are usually enough [5 minutes].
- 4) As a group, they define and list quality attributes of a LEGO build, then choose three of them [5 minutes].
- 5) As a team, they assign points to each LEGO object they have. They have a budget of 30 points to allocate (allocated points must be integers). The students record them on the whiteboard [7 minutes].
- 6) They swap the set of objects with the team on their right hand (Optional).
- 7) As a team, they choose one of the envelopes numbered from 1 to 8. It contains the building process that each team will follow.
- 8) As a team, they organize themselves according to the building process allocated to them [3 minutes].

During the Building-and-Assessment phase, the teams go through several iterations of building an object using the bricks and then assessing the objects built by other teams and allocating points to them based on predefined quality attributes. The steps for this stage are:

- 9) As a team and once the time starts counting, the students build the objects in the following order and one per iteration [10 minutes]:
 - Animal
 - Building
 - Vehicle
 - Object
- 10) As a team and after each iteration, the students go to a different team and ask them to assess the quality of their LEGO build.
- 11) As QAs, they assess the LEGO build using the attributes defined in 4. Then, based on their assessment, they assign points to the LEGO build [3 minutes].
- 12) As a team, the students get the points that the QA allocated to them and record them on the whiteboard.
- 13) As a team, they place the LEGO object on the city plan.
- 14) As a team, they discuss improvements to the building process and apply them in the next iteration [2 minutes].
- 15) The students repeat the steps until they have built all the LEGO objects assigned to the team.

In the end, the Discussion stage is carried out as a group promoting reflection and a wrap-up of the session.

- 16) As a group, the participants add the points up and discover the winner.

objects	teams			
	A	B	C	D
VEHICLE	6/7	1/4		
BUILDING	5/6	13/15		
ANIMAL	2/3	4/7		
OBJECT	10/14	3/4		
total points	23	21		

Fig. 3. Allocated and earned points per object and team.



Fig. 4. City plan at the end of the activity

17) They reflect on the activity and share their experience. Some questions to promote participation are:

- Have you achieved your objective?
- Does your product have good quality?
- Did you come across any challenges when assessing the quality attributes?
- What are the advantages of the building process assigned to you?
- What are its disadvantages?
- How did you manage collaboration and communication issues, if any?
- How would you improve your building process?

Figure 3 presents the board showing the teams (columns), objects (rows), points allocated to each object (green), points earned after the quality assessment is performed (red), and total points (orange) at the end of the activity.

Figure 4 shows the city plan once the last iteration is finished.

D. Building Process Approaches

Each team has to follow a predefined building process to create LEGO objects; there are three different approaches:

Task-oriented approach:

- 1) Analyze the LEGO build requirements.

- 2) Determine and select the necessary bricks.
- 3) Use the bricks to build it.
- 4) Verify if the build corresponds to the ones specified in the requirements.
- 5) Get your build assessed.

Role-oriented approach:

- **Provider:** you have access to the LEGO specifications and you are in charge of selecting the required bricks. You give instructions to the Builder on how to assemble the object.
- **Builder:** you are in charge of building the object using the bricks and instructions given to you by Provider. You do not have access to the LEGO specifications.
- **Guide:** you are in charge of giving instructions to the Builder. You have to verify that the specified constraints are followed, and the object corresponds to the one from the specifications. You have access to the LEGO specifications.
- **Assessor:** you are in charge of assessing the builds created by other teams.

Organize yourselves approach, which allows the team to decide on how they will run their process.

The teams must follow the approach at least during their first iteration, after which they are allowed to adjust and improve their building process for subsequent iterations. That is exactly the moment for the students to put their SPI knowledge in practice.

It is worth mentioning that more approaches can be added; however, in this activity, we decided to play with the rigidity of the process: offering options of the most rigid (task-oriented), a more balanced one (role-oriented), and the most flexible (self-organized).

Lastly, the quality aspect can be traced at different levels of the activity. While it is explicitly addressed as a role (Assessor) in the role-oriented approach, it becomes less visible in the task-oriented approach (step 4), and disappears in the self-organized approach. This decision was made in order to provide students with a wider variety of situations, aiming at inspiring suggestions of improvement and awareness among students during the Discussion phase.

E. SQA and SPI Come into the Game

In order to understand how the activity connects theory and practice, a mapping between the SQA and SPI topics and the learning domains is described below using the cognitive levels of Bloom's taxonomy (L1-Remember, L2-Understand, L3-Apply, L4-Analyze, L5-Evaluate and L6-Creat).

Quality and Process definitions:

- *Remember (L1)* the definition of quality, process and the components of a process (e.g. roles, tasks and products).
- *Recognize (L1)* the process elements present (or absent) in their building process.
- *Understand (L2)* the term quality in terms of the new context.

Students were taught several definitions of quality (e.g. ISO/IEC 25010 [6] and IEEE 730 [7]) and process (e.g. ISO/IEC 12207 [28]).

Process Improvement:

- *Contrast (L2)* the advantages and disadvantages of each building process and its potential improvements.
- *Implement (L3)* the improvements in a timely, effective and efficient manner.
- *Detect (L5)* the nonconformities of their process and product.

As a strategy for guiding their SPI, students were taught the Plan–Do–Check–Act (PDCA) cycle, which is commonly used for continuous improvement of processes and products.

Quality Assurance:

- *Explain (L2)* the criteria to be used for the quality assessment of the products.
- *Execute (L3)* verification and validation procedures for checking that the product meets the specifications.
- *Evaluate (L5)* the quality of the products based on objective criteria.
- *Critique (L5)* their own process and product in order to identify shortfalls and limitations.

During the course, students have been taught the SQA definitions (IEEE 610.12 [8] and IEEE 730) and process (CMMI-DEV [29] and ISO/IEC 12207).

Quality models, attributes and metrics:

- *Recognize (L1)* the most relevant/valuable characteristics of the product.
- *Analyze (L4)* the suitability of the quality models they know to the problem they are solving.
- *Create (L6)* a definition of their own quality attributes and metrics.
- *Adapt (L6)* a known quality model to the current context and needs by providing adjusted quality attributes definitions.

Some examples of the quality attributes defined by the students are:

- **Brick accuracy:** the degree to which the object conforms to the build requirements (color and shape of bricks).
- **Similarity:** the degree to which the object *looks like* the picture in the specification.
- **Playability/Entertainability:** the degree to which the object has attributes that makes it pleasing and satisfying to play with.
- **Structural integrity:** the degree to which the structure of the object is sound and resistant.
- **Building complexity:** the degree to which the object is easy to build.

To participate in the activity in the most advantageous manner, it is expected that the students will bear in mind and apply the ISO/IEC 25010 and the Goal Question Metric (GQM) approach as well as other knowledge previously discussed in class.

TABLE II
SURVEY RESULTS

Statement	Average	Median
The activity is attractive	4.64	5
The way the activity works suits my way of learning	4.16	5
The activity content is connected to other knowledge I already have	4.36	4
While completing each step of the activity, I felt confident that I was learning	3.76	4
I had fun	4.96	5
I lost track of time during the activity	4.72	5
This activity is appropriately challenging for me, the tasks are neither too easy nor too difficult	4.20	5
I would recommend this activity to my peers	4.72	5
I would like to try similar activities in the future	4.68	5
The activity contributed to my learning	4.28	5

F. Data Collection and Analysis

The activity has been applied during three consecutive years with the following number of participants: 10 students in 2018, 27 in 2019 and 31 in 2020. The participants possessed advanced knowledge about software engineering principles, practices and processes from other courses, and had gained industry experience in internships. Fifty-six students identified themselves as males while 12 as females. Most of the students were from New Zealand (59) and the rest were overseas students from seven different nationalities. Lastly, the students’ age median was 22 and the average was 23.0.

After the conclusion of the activity, the students were asked to answer a survey anonymously. The survey consisted of two parts: the first part presented a subset of a standardized questionnaire developed by von Wangenheim et al. [30]; in the second part the students were asked to choose from a list of nine LOs, which they believed were tackled by the activity, followed by their opinion as a free text.

Since 2018, 25 students have answered the survey. Table II presents the average and median for each of the statements of the survey.

Based on the survey results and the students’ opinion, the activity is highly rated in the fun factor (average: 4.96), students seem to enjoy and are engaged throughout the activity. They lose track of time (4.72) and would like to have similar activities in the future (4.68).

Although the activity was very well received, there are aspects on the learning side that could be improved, particularly in regards to emphasizing the learning aspects during the activity (3.76).

In the LOs part of the survey, 78% of the participants perceived they achieved LO1, 67% achieved LO4, with 56% and 33% choosing LO6 and LO2 respectively. Two additional LOs were selected by the participants: LO3 – Hypothesize specific improvements to make the three P’s more effective,

efficient and reliable, and LO5 – Implement and justify quality management systems in a controlled and organized manner, allowing a continuous improvement. Both were selected by 44% of the participants.

Almost no difference in responses of the participants between individual years were observed; however, in 2019 the most number of students, out of those who participated in the activity, answered the survey (16 out of 25).

G. Evolution of the Activity

Over the years, the activity has evolved as based on the pursued learning goals and the target audience, at the same time improving in response to feedback from the participants. The predecessor activity of KUALI-Brick was created in 2012; its motivation was to teach the basic concepts of Kanban (work-in-progress limit, board structure, work flow and team collaboration) to IT professionals differently. The goal of the activity then was to build basic buildings (a school, a hospital, a two-level house, etc.) using bricks, slope bricks and plates (1x1, 2x1, 2x2 and 2x4). In addition to the buildings, the participants had to draw a city plan in an A0 paper sheet using markers. The main condition was that teams had to coordinate their work and communicate through the Kanban board only.

Given the limited variety of bricks and being process-rather than product-focused, the buildings were simple and not so challenging. A couple of years later, the buildings were replaced by sets of ‘2D objects’ (e.g. a ghost, a banana and a watermelon in one set), which increased the challenge level resulting in more fun and engagement for the participants.

In the following few years, the focus of the activity changed from Kanban to SPI, where the participants discussed improvements and adapted their process and workflow after each iteration. With the aim of making the activity more competitive, another step was added, in which teams had to ‘sell’ their builds to other teams in order to gain money. In the end, this ‘selling’ step was formalized as a quality assessment and SQA concepts were added into the mix. Since the focus shifted from Kanban to SPI and SQA, the Kanban board was removed, the bricks variety increased and the types of builds became more complex.

V. DISCUSSION

The discussion is organized in four subsections: positive aspects, negative aspects, lessons learned and limitations of the study.

A. Positive Aspects

As reported by most of the LEGO-related studies presented in Section II, participants experience an increase in engagement and enjoyment.

We have observed an overwhelming evidence that the fun level of the students rockets as fun has obtained the highest score (4.96) in the survey. In addition, students have commented that the activity was the “*most fun lecture I’ve had in 3.5 years*”, “*loved the activity, thank you for putting it together*”.

The engagement of the participants is very high: “*it was really engaging, and got us to think about to topic in a context other than software which I think allows for different perspectives*”.

The activity breaks the routine and makes the time more enjoyable: “*this was one of the best activities I’ve done at uni*”. The playful nature of the activity positively influences the attitude of the participants towards the learning process: “*it was a fun and memorable way of learning the concepts*”.

Its hands-on approach gives the opportunity to put theoretical concepts into practice: “*I thought it was a fantastic way of putting into practice some of the concepts we’ve learned in class*”.

The fact that students, being used to construct ‘abstract’ software, can see the palpable objects they have build, facilitates acquisition of such abstract concepts as quality, and their transformation into concrete applications. “*It helped apply some of the concepts that I had learned from the course into a game like scenario which is more of a concrete structure than the abstract nature of software*”. As a consequence, the participants realize how a quality attribute or metric affects the assessment process of a real product. They also become aware of how a timely adjustment can improve the building process by making it more efficient or by enhancing the quality of the final outcome.

B. Negative Aspects

Stress associated with short iterations [18] and time constraints have been observed in other studies. In this experience, time was also a topic of concern. Not having enough time for building the objects and/or discussing at the end of the activity were the main issues expressed by participants. “*Due to the overrun in time restricting the opportunity for reflection and discussion about the relation to software process and quality*”. It was observed that the lack of a sufficient discussion time creates difficulty in connecting theory and practice.

As mentioned before, the set-up of the room is quite important. In this experience, in 2018 the boxes with bricks were sitting on a table on one side of the room, giving participants only two sides of the table to reach the bricks. For subsequent instances, the boxes were located in a table in the middle of the room, drastically improving access to them: “*A few things just need to be ironed out like the mosh pit trying to get lego and the timings of things*”.

Another constraint might be the price of the game material; however, the output is worth the investment. The cost of the bricks, boxes and the city plan printing was approximately \$1,000 NZD (\$715 USD).

C. Lessons Learned

The discussion at the end of the activity is fundamental in order to emphasize the learning outcomes. Participants lose track of time easily and at early stages of the activity. In addition, the time allocated is tight, therefore it is most likely that the discussion at the end of the activity might be affected due to time constraints. In the 2019 instance of the activity,

the discussion was shortened from 20 to 5 minutes, causing disagreement expressed by some students. On the contrary, when the discussion is carried out as planned, the learning aspects are boosted – demonstrated by a comment from the 2020 instance: “*The discussion guided by Miguel about how the activity was *really* valuable and it really helps to relate what we did to what we have been learning*”.

Prepare students for the activity, let them know what will happen. For the 2019 instance, students were informed about the activity with no major details provided, which was supposed to be a ‘surprise’. It might not be an issue due to the fact that the activity is neither formally assessed nor requires preparation from students; however, some students can perceive it differently: “*it is very different to normal activities undertaken in class and was somewhat of a shock when I had come prepared to take notes and the like*”. In order to avoid undesirable surprises, we included the activity into the course schedule and it is advisable to remind students of the activity once again.

Make the most of the fun boost. In order to make the most of the activity, we suggest conducting it as an optional activity of the course; this motivates the students to have the right attitude. The impact of the activity is important, keeping the momentum going in the participants long afterwards and outside the course content; this was observed from 2019 and 2020 students: “*It was so fun, we definitely needed an activity like with all the work we’ve been putting towards the end of the semester with all the deadlines around us*”; and “*It was certainly a welcome activity to have some fun during a very stressful period of time. Please keep running this!*”.

Aesthetics are important. Such aspects as the set-up of the room, the bricks in boxes organized by colour and size, and the city plan have a positive effect on students. From the beginning the students enter the room, the experience should be welcoming and making them feel that the lecture material was specially prepared for them – this breaks the routine and boosts engagement, which is transferable to following lectures.

Custom vs. Packaged LEGO sets. On the one hand, custom sets are more difficult to build and assess as they introduce extra variables into the activity making it more complex. For example, participants have to rely on their creativity and ability to build, so the time becomes an issue. Packaged sets, on the other hand, offer a clear image of what the final product should look like, at the same time reducing the creativity part. We suggest avoiding branded sets (e.g. Star Wars or Friends) and using the LEGO Classics ideas that contain small sets and ideas to build with no limits to creativity. Up to now, the Classics set has brought the better results in our experience.

Add variability to the final iteration. After the second iteration participants might perceive the dynamic as repetitive, therefore we suggest addition of some variability to the third iteration. The simplest and most effective variation in our experience was asking the teams to build two instead of one objects during the third iteration. With the time constraints in place, the participants are forced to rush their process causing

the quality of the products to be affected. We call this variation “On a rush!”. Another suggestion is to shuffle team members around the teams in order to make the process adjustments more challenging.

D. Limitations

The main limitation of this work is the lack of a formal assessment to confirm improvement in the participants’ knowledge. Currently, the claim of improvement is based solely on observations and participants’ opinions.

Secondly, while the activity allows the students to experience, relate and reflect on quality attributes, metrics, SQA and SPI issues, they are exposed to simplified versions of those concepts only. As stated by [24], the LEGO metaphor allows to simulate some characteristics of software development, but it is not a substitute for actual software development tasks.

Lastly, if the playful aspect is not controlled adequately, the activity can turn into a chaos, given the euphoria of participants and the relaxed environment in which it is carried out. Here, the instructor’s experience plays an important role in controlling the dynamics of the class and in keeping the participants focused on the learning goals.

VI. CONCLUSIONS AND FUTURE WORK

While knowledge acquisition is one of the most important aspects in every learning context, there are other relevant aspects that help creating an effective learning environment in a lecture room. The attitude towards the lecturer, other classmates and the course content are variables that affect the student performance and achievement of the learning outcomes. The fun aspect is an unexpected attribute during a fourth-year Engineering lecture; however, adding it to the mix can affect the student attitude in a positive way.

The KUALI-Brick activity was created with the aim of involving students into application of theoretical concepts and giving them the opportunity to experience those concepts firsthand. We can conclude that the KUALI-Brick activity resulted in an excellent tool for making fourth-year Software Engineering students aware of how abstract concepts can be instantiated in concrete objects.

In general, we can state that the research around the combination of LEGO and education in Software Engineering is promising. This mix almost secures a superb learning experience in terms of fun, engagement and enjoyment, but still needs evidence to become recognized as a game that delivers quantitative improvement in the learning outcomes of the participants.

As future work, an assessment item will be created to evaluate the learning aspect of the activity. Besides, the KUALI-Brick activity rules and steps will be made available in order to share and obtain feedback from the community.

ACKNOWLEDGMENT

The author would like to thank Polina Maslova for her contribution to the activity design.

REFERENCES

- [1] P. Crosby, *Quality Is Free: The Art of Making Quality Certain*. New American Library, 1979.
- [2] W. S. Humphrey, *Managing the Software Process*. Addison-Wesley Longman Publishing Co., Inc., 1989.
- [3] S. H. Kan, *Metrics and Models in Software Quality Engineering*. Addison-Wesley Professional, 2003.
- [4] S. Pfleeger and B. Kitchenham, "Software Quality: The Elusive Target," *IEEE Software*, pp. 12–21, 1996.
- [5] A. Gillies, *Software Quality: Theory and Management*. Lulu.com, 2011.
- [6] ISO/IEC, *25010:2011 Systems and software engineering – Systems and Software Quality Requirements and Evaluation (SQuARE) – Systems and software Quality Models*. ISO/IEC, 2011.
- [7] IEEE, *730-2014 Software Quality Assurance Processes*. IEEE, 2014.
- [8] —, *610.12-1990 Glossary of Software Engineering Terminology*. IEEE, 1990.
- [9] M. T. Baldassarre, M. Piattini, F. J. Pino, and G. Visaggio, "Comparing ISO/IEC 12207 and CMMI-Dev: Towards a Mapping of ISO/IEC 15504-7," in *2009 ICSE Workshop on Software Quality*. IEEE, 2009, pp. 59–64.
- [10] S. Kurkovsky, S. Ludi, and L. Clark, "Active Learning with LEGO for Software Requirements," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 2019, pp. 218–224.
- [11] T. D. Lynch, M. Herold, J. Bolinger, S. Deshpande, T. Bihari, J. Ramanathan, and R. Ramnath, "An Agile Boot Camp: Using a LEGO®-based Active Game to Ground Agile Development Principles," in *2011 Frontiers in Education Conference (FIE)*. IEEE, 2011, pp. FIH–1.
- [12] D. A. B. Bica and C. A. G. da Silva, "Learning Process of Agile Scrum Methodology with Lego Blocks in Interactive Academic Games: Viewpoint of Students," *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, vol. 15, no. 2, pp. 95–104, 2020.
- [13] J.-P. Steghöfer, E. Knauss, E. Alégroth, I. Hammouda, H. Burden, and M. Ericsson, "Teaching Agile-Addressing the Conflict Between Project Delivery and Application of Agile Methods," in *2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)*. IEEE, 2016, pp. 303–312.
- [14] M. Paasivaara, V. Heikkilä, C. Lassenius, and T. Toivola, "Teaching Students Scrum Using LEGO Blocks," in *Companion Proceedings of the 36th International Conference on Software Engineering*, 2014, pp. 382–391.
- [15] D. Lübke and K. Schneider, "Agile Hour: Teaching XP Skills to Students and IT Professionals," in *International Conference on Product Focused Software Process Improvement*. Springer, 2005, pp. 517–529.
- [24] A. Šablis, J. Gonzalez-Huerta, E. Zabardast, and D. Šmite, "Building LEGO Towers: An Exercise for Teaching the Challenges of Global Work," *ACM Transactions on Computing Education (TOCE)*, vol. 19, no. 2, pp. 1–32, 2019.
- [16] S. Kurkovsky, "A Lego-based Approach to Introducing Test-Driven Development," in *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, 2016, pp. 246–247.
- [17] J.-P. Steghöfer, "Using Essence in a Software Development Methodologies Course: An Experience Report," in *Software Engineering (Workshops)*, 2019, pp. 7–10.
- [18] J.-P. Steghöfer, "Providing a Baseline in Software Process Improvement Education with Lego Scrum Simulations," in *2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*. IEEE, 2018, pp. 126–135.
- [19] S. Kurkovsky, "Teaching Software Engineering with LEGO Serious Play," in *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, 2015, pp. 213–218.
- [20] H. Mayr, "Teaching Better Requirements Engineering Using LEGO® Serious Play™," in *Proceedings of The 9th International Conference on Society and Information Technologies*. IIS, 2018, pp. 126–131.
- [21] K. Gama, "An Experience Report on Using LEGO-based Activities in a Software Engineering Course," in *Proceedings of the XXXIII Brazilian Symposium on Software Engineering*, 2019, pp. 289–298.
- [22] D. J. Hood and C. S. Hood, "Teaching Software Project Management Using Simulations," *ACM SIGCSE Bulletin*, vol. 38, no. 3, pp. 289–293, 2006.
- [23] S. Kurkovsky, "Using LEGO to Teach Software Interfaces and Integration," in *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 2018, pp. 371–372.
- [25] L. M. Laird and Y. Yang, "Engaging Software Estimation Education Using Legos: A Case Study," in *2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)*. IEEE, 2016, pp. 511–517.
- [26] S. Kurkovsky and S. Ludi. (2021) LEGO for Software Engineering. [Online]. Available: <https://web.ccsu.edu/LEGO-SE/>
- [27] CSSE University of Canterbury. (2020) Software Process and Product Quality. [Online]. Available: [https://www.canterbury.ac.nz/courseinfo/GetCourseDetails.aspx?course=SENG403&occurrence=20S2\(C\)&year=2020](https://www.canterbury.ac.nz/courseinfo/GetCourseDetails.aspx?course=SENG403&occurrence=20S2(C)&year=2020)
- [28] ISO/IEC, *12207:2008 Systems and software engineering – Software life cycle processes*. ISO/IEC, 2008.
- [29] SEI, *CMMI for Development (CMMI-DEV), Version 1.3*. Carnegie Mellon, 2010.
- [30] C. G. von Wangenheim, R. Savi, and A. F. Borgatto, "DELIVER!–An Educational Game for Teaching Earned Value Management in Computing Courses," *Information and software Technology*, vol. 54, no. 3, pp. 286–298, 2012.