Experience Report on Teaching Testing through Gamification

Beáta Lőrincz
beata.lorincz@ubbcluj.ro
Babes-Bolyai University
Faculty of Mathematics and Computer Science
Romania

Bogdan Iudean
bogdan.iudean@gmail.com
Babes-Bolyai University
Faculty of Mathematics and Computer Science
Romania

Andreea Vescan
andreea.vescan@ubbcluj.ro
Babes-Bolyai University
Faculty of Mathematics and Computer Science
Romania

ABSTRACT

Software systems become increasingly complex and testing is a vital component of the development process. Teaching software testing concepts are now more than ever acknowledged as essential.

The aim of this paper is to report on our software testing teaching approach, using game-based activities to engage students and facilitate learning, making them aware of their actions and related testing concepts. Agile testing and Session-Based Test Management are learned through lego-based context, exploratory testing is learned through a dice-based game, and various testing concepts are learned during the laboratory using storification.

We report on the results of activities with students, extracting valuable lessons for reproducing this approach in teaching software testing: game-based learning motivated students to participate in the activities, reflection on their actions allowed them to self-discover the testing concepts encapsulated into the game. In addition, we adapt and analyse an industry-like environment that serves as experience for their future careers.

CCS CONCEPTS

• **Software and its engineering → Software testing and debugging**; **Use cases**; • **Applied computing → Interactive learning environments**; **Collaborative learning**.

KEYWORDS

Software Verification and validation, Software testing, Agile testing, Exploratory testing, Experiential learning, Gamification, Storification

ACM Reference Format:
Beáta Lőrincz, Bogdan Iudean, and Andreea Vescan. 2021. Experience Report on Teaching Testing through Gamification. In *Proceedings of the 3rd International Workshop on Education through Advanced Software Engineering and Artificial Intelligence (EASEAI '21), August 23, 2021, Athens, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3472673.3473960

EASEAI '21, August 23, 2021, Athens, Greece

ACM ISBN 978-1-4503-8624-1/21/08…$15.00
https://doi.org/10.1145/3472673.3473960

1 INTRODUCTION

Learning is essential to our existence, being continuous through the life of any individual. Information and continuous learning nourishes our minds, being part of the personal and professional development in an effort to avoid stagnation and to reach our full potential.

The learning studies [25] throughout the last decades revealed that the students no longer want to play a passive role in their learning process, but rather prefer to have an active role. Mayer [26] advocates that learning is a process of knowledge construction, thus students interpret the received information. David Kolb [21] describes learning as following a cycle of stages, being directly connected to the way the student processes experiences and later reflects upon those experiences.

The Computing Curricula 2020 (CC2020) [9] proposes a competency based characterization of computing organizing sub-disciplines using competencies. Verification and Validation in Software Engineering has a level of minimum 3 to maximum 5 for undergraduate (computing knowledge). The main competences in software testing that are mentioned in the Software Engineering competences refer to performing an integrative test and analysis of software components by using black-box, conducting a regressive test of software components, conducting a test utilizing appropriate testing tools, and planning and conducting processes to design test cases.

Engaging and motivating students about testing software systems is especially challenging (at our university being part of the last semester), and without the awareness of the impact of the subject in real-life software development, the topic of software testing seems for many of them less important or exciting.

Teaching testing related concepts requires various approaches from active learning [6] to Lego [4], and to Gamification in education [23]. Additional studies also focus on engagement and staying in the flow [17].

The aim of this paper is to report on the results of activities with students regarding teaching software testing using game-based activities in all three teaching/learning contexts: lectures, laboratories and seminars. We employ lego-based game for the agile testing concept, storification for the laboratory setting with various testing concepts to be learned, and dice-based game for learning exploratory testing in the seminar setting.

The rest of the paper is organized as follows: Section 2 briefly presents the concepts to be learned and the learning methods. The methodology (research questions, course details, activities) is outlined in Section 3. The results are presented in Section 4, emphasizing the students perception and learning outcome, along with the answers to the research questions. Section 5 outlines several lessons learned, and Section 6 states the outcome of our experience.

2 BACKGROUND

This section details the main ingredients of our teaching/learning context, first, about the concepts to be learned from the software testing domain (Agile testing, Exploratory testing, various testing concepts), and second about the learning methodology (experiential learning, game-based learning, and using metaphors).

2.1 Learned Concepts - Testing

An overview of the taught concepts that are investigated in this study are provided in what follows.

2.1.1 Agile Testing. In today's software development, agile methodologies are most frequently used for managing projects and organizational culture within companies [12]. The agile approach splits the development process into iterations and accepts, responds to changes [31]. Compared with traditional methods that have rigid boundaries agile is a feasible approach for dealing with high uncertainty. As stated in The Agile Manifesto [14] it is more effective to accommodate changes as opposed to preventing them.

Agile testing strategies refer to incorporating testing throughout the full life-cycle of the project [3]. The iterations called sprints in the development process include testing activities such as unit and functional testing that require as a final step the approval of the product owner. The sprint planning sessions define the QA team activities as well, such as the scenarios selected for manual, automated, or performance testing.

The Session-Based Test Management (SBTM) [5] method proposed by Jonathan Bach follows an organized and orderly way of working. The essence of this technique lies in the results at the end of the session (notes, issues, bugs discovered) and in the discussion between the leader of the test team and the tester.

2.1.2 Exploratory Testing. The "exploratory testing" term was first coined by Cem Kaner in his book [19], as "a style of software testing than emphasizes the personal freedom and responsibility of the individual tester to continually optimize the value of his/her work". James Bach provides in his paper [33] the following definition: "exploratory testing is simultaneous learning, test design, and test execution".

Hendrickson [16] details various aspects of exploratory testing, from chartering the exploration to adding various dimensions, i.e., varying sequencing and interactions, entities relationships, state and transitions, interfaces and exploring requirements. Thus, there is less of a structure and a specified process, the tester having more personal freedom and responsibility to use their knowledge to uncover software issues which are unanticipated. The key elements for exploratory testing are: charters that act as a guideline, sessions that are time bound, and mind maps that define the testing goals capturing the relationships between applied tests, features and findings during testing. Exploratory testing is well-suited for the agile development approach as it is more adaptable to changes.

2.1.3 Various Testing Concepts. The taught testing concepts include: the inspection/review process, the design of test cases using black-box, white-box testing approaches, different levels of integration testing, automated web UI testing and software correctness:

- During the inspection phase the requirement, design documents and source code are reviewed using checklists.
- The test case design approached by black-box testing allows for focusing on the system's functionalities without examining the internal structure of the source code, while the white-box testing draws the attention to covering the possible paths of the source code.
- Integration testing is a vital component of software testing that is indispensable for testing software systems as a whole.
- The web testing techniques frequently used in today's software development provide an insight for controlling website functionalities.
- The perspective of software correctness is especially useful for safety-critical systems.

2.2 Learning Methodology - Pedagogy

An overview of the used learning elements are provided next, detailing the experiential learning theory and the use of metaphors in teaching/learning along with storification and gamification.

2.2.1 Experiential Learning Theory. Kolb [21] proposed the Experiential Learning Theory (ELT) having the goal to incorporate common themes in the work of the "foundational scholars" of experiential learning into a systematic framework that can depict and advance the problems of learning and education, i.e. a perspective of the learning process that could be applied " in all situations and areas of life".

Figure 1 graphically presents the learning cycle [21]: the experience (Concrete Experience –CE) is translated through reflection (Reflective Observation –RO) into concepts (Abstract Conceptualization – AC), which in turn are used as guides for active experimentation and the choice of new experiences (Active Experimentation – AE).

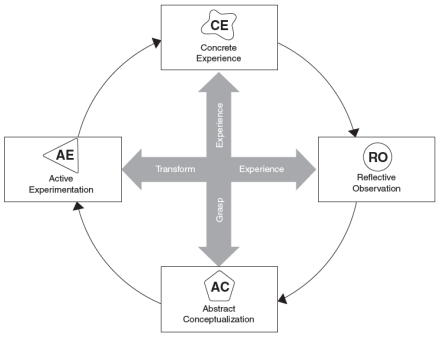


Figure 1: Kolb's Learning cycle [21]

There are various Experiential Learning (EL) studies in various fields, the learning approach [20] being applied in 30 different professions and academic disciplines: chemical engineering education [1], information and communication technology [8], applications of mathematics in practice [30], computer science [34].

2.2.2 Gamification/Metaphors in Teaching/Learning. A short introduction about learning styles, storification and gamification is provided in what follows.

Visual and narrative learning styles. According to Curry's "Onion" Model presented in the Handbook of Intellectual Styles [35], students are more likely to learn the subject when a matching mode of teaching based on their preferred mode of learning comes into effect. Byrne stated in [29] that up to 40% of students are learning best when the presented tool is multimodal, allowing different approaches based on interest. As stated by Mayer in his applied study of learning in [13], people encode information more efficiently when animation and narrations are presented at the same time. Based on his previous research from [27] visualizing interactive symbolic images in conjunction with auditory explanations provided more successful learning expectations. Using this method has also reduced the risk of any distraction, and it led students to pay more attention to the conceptual abstraction.

Storification. Storification has its roots in the abstraction of palpable concepts into metaphorical events. Converting traditional concepts by giving them motion throughout characters, environments and original scenarios can boost the students' focus, engagement, and understanding. As a story can be told through a handful of communication channels besides the traditional narration, such as music, dance, roleplay, and drawings, this ensures a bigger chance in fitting the learning style of the student. As a story has a consecrated structure and traditional elements such as heroes, villains that cause the conflict, moments of hardship for the protagonists and unexpected help from unexpected allies, the shame should be present among the stories from a lecture. This creates a feeling of familiarity and ensures the fact that the concepts are no longer just static blocks of information, they gain motion, context and, most importantly, meaning.

Gamification. The proposed definition of gamification by [10] defines it as "the use of game design elements in non-game contexts". The application of gamification elements has particularly gained interest in the domain of education where preserving the motivation and engagement of students is a constant challenge [24]. Several studies investigate the effects of employing gamification elements in education [2, 7, 11] and report positive effects of gamified education. Gamification elements embodied into the learning processes typically include achievements/badges, rewards, storification, points, leaderboards, levels and challenges [15]. The affordances of the implications of these elements are not only measured quantitatively (e.g. engagement of students) but the psychological and behavioural outcomes are also considered (e.g. motivation, enjoyment). Review studies that aggregate the findings of gamified experiments and target the question whether gamification works [15, 24] conclude with mainly positive results, but suggest that contextual factors, design elements, personality, and demographic aspects need to be considered when implementing gamified solutions. Negative outcomes reported included elevated competition and evaluation difficulties.

3 METHODOLOGY

The objective of this research is to investigate and report on the use of various game-based activities for learning software testing concepts. We aim to investigate at different levels of teaching (lectures, seminars, laboratories) how learning is best achieved by using various game-based activities (lego - based, dice - based and metaphor). Figure 2 depicts the overview of our investigation: learning software testing concepts (testing using Agile or Session Based Test Management methodologies, Exploratory testing, Inspection, Test case design, Levels of testing, etc.) by activities in all aspects of teaching/learning (lecture, seminar, laboratory) using various games.

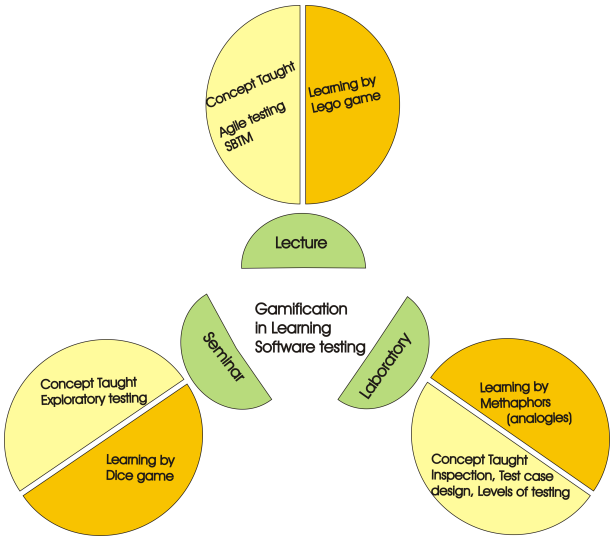


Figure 2: Research design overview

Firstly, we emphasize the research questions, their details are provided in the next section.

3.1 Research Question

The objective of this study is next formulated into several research questions targeting various aspects of learning.

RQ1: Is Gamification effective in teaching Software testing?

RQ2: What is the impact of using "Gamification" on the students engagement and motivation to actively participate in the activities?

RQ3: What are the students perceptions and emotions regarding the game-based software testing learning activities?

The direct observations on the students' feedback on the lecture activity, discussion during the activity and the analysis of the team results are used to answer RQ1.

The feedback of each step of the game-based learning and a survey that the students answered were used to answer RQ2 and RQ3.

3.2 Course and Participants

The course under analysis is set up as an undergraduate level, i.e. in the "Computer Science" program (third year of study). The title of the course is "Software System Verification and Validation" (SSVV). The course is taught during 12 weeks, having 2 hours for lecture, 1 hour for seminar work (in class and take home), and 1 hour for

laboratory work (in class and take home) with dedicated hours for home study (project hours). The taught verification and validation concepts are: inspection/review, design the test cases using black-box testing and white-box testing techniques, level of testing (various integration strategies), exploratory testing, UI web testing (Selenium), symbolic execution, model checking, correctness (Floyd, Hoare, Dijsktra, Correctness-by-Construction). The majority of the introduced concepts are new to the students, with the exception of test case design that is part of previously thought subjects' curricula as well. The seminar and laboratory activities are grouped and concentrated on several aspects of the taught concepts, in what follows we will only describe the one that we analyse in this paper.

3.3 Description of Activities

The current section describes for each teaching activity (lecture, seminar, laboratory) only the taught testing concepts for which game-based approaches were adopted, along with steps and aims of the activities. It is important to state that the taught concepts in the course of lecture and seminar activities are then used during laboratory activities, the students being thus able to practice the learned concepts.

3.3.1 Activities at Lectures. One selected lecture was dedicated to agile testing and a gamified experiment for finding faults. The structure of the lecture was designed to have both a theoretical and an experiential learning phase. The lecture started off with two invited speakers with experience in the software development industry as project managers and scrum masters. After presenting basic concepts related to agile development and testing processes, the speakers shared their experiences, challenges and lessons learned from their career paths.

After the theoretical presentation and discussion, the students were presented with pictures and videos of a Lego house and the problem of accommodating a family of five members (parents, children including a toddler). The pictures taken from several angles about both the family and the house, and the video capturing the same information provided a detailed overview of the problem at hand. In Figure 3 the lego house is presented from one angle.

Figure 3: Lecture Agile testing and the Lego House

The gamified experiment consisted of investigating whether the house is suitable for rent for the family. 37 students were split into 5 groups, with the task of conducting a SBTM activity to determine the appropriateness of the house for the selected family.

The activity followed the guidelines of SBTM in an informal manner with students starting off with chartering, followed by an uninterrupted testing time and taking notes, reporting. After a 25 minutes session the findings/bug listings were presented by each of the groups. Students were asked to vote which team they considered the most successful in finding the faults and to provide feedback on the lessons learned.

3.3.2 Activities at Laboratories. The laboratory is structured into a handful of modules that are tackling a bottom-up strategy. The students are incrementally guided through the various stages of software validation and verification implied among the flow of a project from idea to production:

- Exercising the attention to the potential problems, vulnerabilities, and misunderstandings that may occur during the requirements engineering, design and legacy code exploring phases.
- Adapt by using black-box testing and white-box testing strategies based on the degree of program knowledge and transparency.
- Create functional pipelines using Maven, Git, Jenkins, and Testlink to automatically validate changes.
- Use integration testing as a means of ensuring the quality of parallelly developed modules.
- Lay the grounds for automation testing by performing web-based UI testing using the Selenium Framework.

Storytelling

The novelty of the laboratory stands in the storification of the concept. A wide setting is being provided to the students, namely, VVSS in the Stone Age. As the road to the Stone Factory is long and filled with danger due to dinosaurs, the clients raise the requirements for the first car in history. Having the initial problem in place, as in a fairytale, the world gets populated with a handful of characters that will solve it, each of them representing a role in a company or even an abstraction of a technology: Uga and Buga as developers, Tudor the Tester, Unit the Testing Robot, Mecha the Automation Monkey, etc. With a conceptual space, a conceptual project and conceptual characters, the students are exploring the fundamentals of the testing discipline by following the adventures of colorful, meaningful, and entertaining metaphorical caricatures. See Figure 4 for the black-box testing Laboratory.

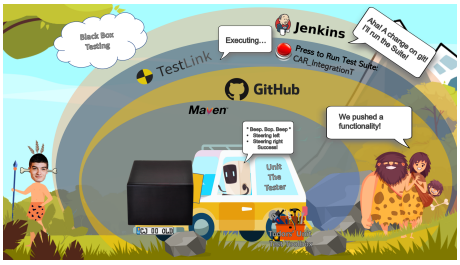


Figure 4: Laboratory - Black-box testing

By holding 10 visual and narrative presentations at the beginning of each class, the following advantages can be noticed:

- Passive learning is more likely to occur as more stimuli are being targeted.
- The students are more engaged as the curiosity and amusement factors come into play.
- The students have an abstract starting-point example which they can later extrapolate to complete their assignments.

The following risks can also not be neglected:

- The chance to misinterpret the metaphor in a potential absence of more concrete examples.
- The chance for this type of presenting the information to not be suitable for all the students based on their personal learning style.

Remark. Students are split into groups for the laboratory activities, and two of the groups were selected to participate in the laboratory activities using storification. Further analysis of the results will be conducted by comparing the performances of these groups with the other groups learning the same concepts introduced in a way that follows the agile principle, thus creating an industry-like environment for the learning processes.

3.3.3 Activities at Seminars. One of the seminars had as a topic *exploratory testing* and in order to experience the process in a fun way, the activity consisted in a dice - based game [28] in which the students had to guess the computation formula, i.e. for a given three dice configurations an associated number was provided. The students could do one of the tasks: (1) roll the dices again, (2) ask a direct question, or (3) guess the formula. The formula involved only "+", "-" or "*" with the values of the dice but had also conditionals like "only even values of dice are considered" or "some dice was not considered at all". For each group we used a different formula, having a different configuration of the dice that affects the formula and varying the odd/even values of dices. For example, in Figure 5 for the three rolled dice, we applied the formula: *FirstEven - LastOdd*, thus the second dice did not "participate" in the formula.

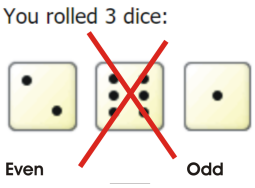


Figure 5: Exploratory testing with Dice formula

The activity was performed using experiential learning steps, after guessing the formula (the Concrete Experience step) each group participated in the next steps: Reflective Observation in which they reflect on the actions they did (" What did you notice about what you did?"), Abstract Conceptualization ("What do you think the intention behind that behaviour was?") in which they find reasons behind their actions, identifying important aspects about exploratory testing, and Active Experimentation ("What is that and how would you start to use that new insight?") where they pinpoint new contexts in which they will be using the exploratory testing elements.

4 ANALYSIS OF RESULTS

This section presents an analysis of the results of the investigation, detailing the elements for each teaching activity (lecture, seminar, laboratory) and the results for the research questions.

4.1 Results for each type of Activity

In what follows we present the obtained results for each considered activity, i.e. lectures, laboratories and seminars.

4.1.1 Results for Lectures. The gamified experiment's results were presented to each of the participating student groups through a list of bugs they have identified. Several faults were found by each group (e.g., not enough bathrooms and bedrooms), but each group listed issues that were not discovered by other groups (e.g. no pillar for the balcony, no TV) and some of the groups considered aspects outside the scope of the suitability of the house (e.g. no garage). Thus, during the investigation process some students also made connections to requirements.

The experiment followed by a discussion revealed the analogy between the structure of the house and the source code to be tested, the investigation of each room to modules and functions to be verified and the user needs represented by the family members.

The lessons learned were presented to students in the form of a word cloud, and the results of the rankings based on their votes were displayed. The students were also provided with a feedback listing the common bugs identified by all of the teams, and the differences, emphasizing the importance of different views with various experiences, and seeing different perspectives. Some groups have considered future events such as the children growing providing a good example on how the "application" could be used in other contexts or on how it could respond to changing requirements. There were no questions raised about the family, while the feedback revealed the fact that the family was expecting another child. This would have been an important aspect to consider and affirms the importance of communication, especially in the case of agile development where the requirements change at a fast pace.

At the end of the lecture, students were asked to state the concepts that they learned during the game-based lecture, and these were gathered using a word cloud. The students provided a word such as: agile, scrum team, sbtm, testing, session testing, explorative, experience-based testing, team management, thus all the words related to the concept that were facilitated. Students were also asked about their perception of the lecture being organized as a lego - based learning, the received feedback is provided in Figure 6.

Figure 6: Agile Lecture with Feedback by students

Thus, some of the adjectives of the lecture are: interactive, fun, cool, but also interpersonal, social, teamwork, and creative.

4.1.2 Results for Laboratories. At the end of the didactic activity, students were requested to provide feedback regarding their opinion related to the storytelling activities in the labs. Among the highlighted answers, they have reported that the activity had a couple of adjacent benefits aside from the actual learning process. They have reported that it helped them with the following:

- They attended the lab out of sheer curiosity for the story outcome even when they were supposed to be absent.
- They were able to tackle the laboratory problems with ease every time as they had an engaging example presented to them beforehand.
- They signaled that they understood where the concepts are applicable in a production environment and not only in a theoretical context.
- They remembered the information under pressure (for example, during interviews).

The main negative feedback is raising the following risks:

- If students were missing the beginning of the story, it was not easy to adapt or understand its content or meaning on the go.
- If students were misinterpreting the metaphor then it was misleading them by solving the lab problems in a wrong manner.

4.1.3 Results for Seminars. Each experiential learning step with all groups were documented using Mentimeter [32] and analysed by sentiment analysis approaches. The perception of the students about the activity was also investigated by the use of a survey and for some of the open questions sentiment analysis approaches were also used.

Student's perception in each EL phase using their own words.

The obtained results for each seminar activity are available here [22] and for one group we have added in Figure 7 the "words" provided by the students for each experiential learning step. In the *Reflective* step the students felt confused but also focused, competitive, discovering patters, happy with others ideas. In the *Conceptualization* phase they identified that concepts related to *exploratory testing* are: being curious, teamwork, patterns, patience, out of the box ideas.

Student's perception in each EL phase using Sentiment Analysis.

The sentiment of the answers for each experiential learning step was also analysed using sentiment analysis to identify the emotional meaning of the words used by the students. Firstly, they were evaluated using rule-based methods (RBM) with the Sentiment Analysis Python package[1] that relies on the model presented in [18]. The assessment of the sentiments is determined with a set of lexical features extracted from social media texts combined with a simple set of rules that incorporate syntactical and grammatical aspects of the texts analysed. The sentiment of the answers were rated as `Positive`, `Negative` or `Neutral`.

The second evaluation used an attention-based transformer neural network (NNT) to classify answers as `Positive` or `Negative`. The Huggingface implementation[2] is used.

Figure 8a) and Figure 8b) contain the results of the conducted sentiment analysis. The first evaluation resulted in a predominantly neural classification for the answers. The second evaluation that used only positive and negative categories is closer to human interpretation, as words that have a positive orientation (e.g., understanding, teamwork, included, helped) ended up in the positive sphere.

Student's perception in each EL phase using a Survey.

In order to understand better the effect of teaching exploratory testing in a game-based setting with the experience learning method, students were also asked to answer several questions in a questionnaire with 13 questions, 12 that used Likert scale (1-Strongly Disagree, 2- Disagree, 3-Neutral, 4-Agree, 5-Strongly Agree) and two open questions. The number of respondents was 74, 24 women, 49 men, and 1 preferred not to say, with ages between 20 to 23.

The results are available at this figshare link [22] and we outline here the main obtained results regarding each individual steps.

Students were asked if they thought that the Experiential learning was integral (essential, fundamental, necessary) to the seminar content. Among the students, 39 Strongly Agree, 27 responded with Agree, 5 were Neutral and only 3 answered with Disagree.

Regarding the question whether the Experiential learning created better understanding of how to conduct Exploratory testing, the following results were obtained: 46 Strongly Agree, 20 responded with Agree, 7 being Neutral and only 1 answered with Disagree.

For the question regarding whether the seminar activity provided the student with enough work on the topic so he/she could form his/her own conclusions, the following results were obtained: 27 Strongly Agree, 38 responded with Agree, 6 were Neutral and only 3 answered with Disagree.

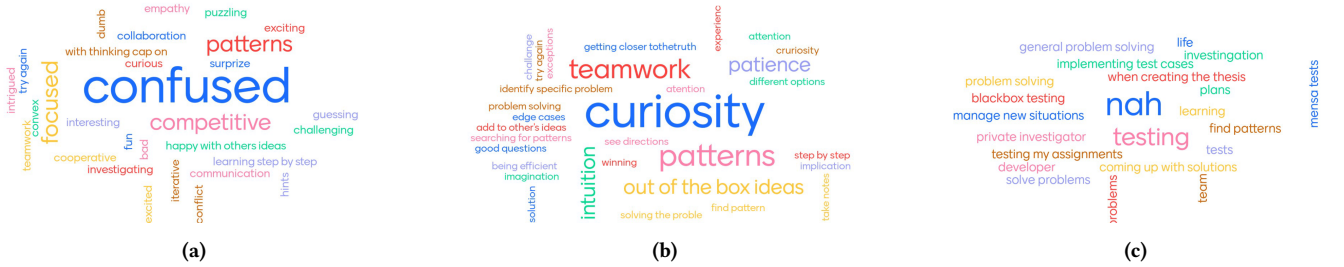
Figure 8c) presents the distribution of the perception of the respondents for each step of the experience learning used in teaching exploratory testing. As seen in the figure, for all steps the majority of respondents answered with Strongly agree and Agree, few being neutral and fewer disagreeing. Thus, the perception of students is that each step of the experience learning positively influenced the learning outcome. See more details here [22].

For the question that targeted if "Experience learning was personally satisfying", the following results were received: 34 Strongly Agree, 27 responded with Agree, 8 being Neutral, 4 answered with Disagree, and 1 answered with Strongly Disagree.

Two open questions were also addressed: "Your benefit or take-away from the Experiential learning experience was ..." and "If you could have changed one thing about your Experiential learning experience, it would have been ...". Sentiment analysis (RBM and NNT approaches) was also applied and the obtained results are provided next, interpreting the overall positive experience and room for improvement: Take-away (NNT - Positive 80% and for Negative 20%; RBM - Positive 53%, Neutral 40%, Negative 7%;), and Improvement (NNT - Positive 20% and Negative 80%; RBM - Positive 28%, Neutral 57%, Negative 15%).

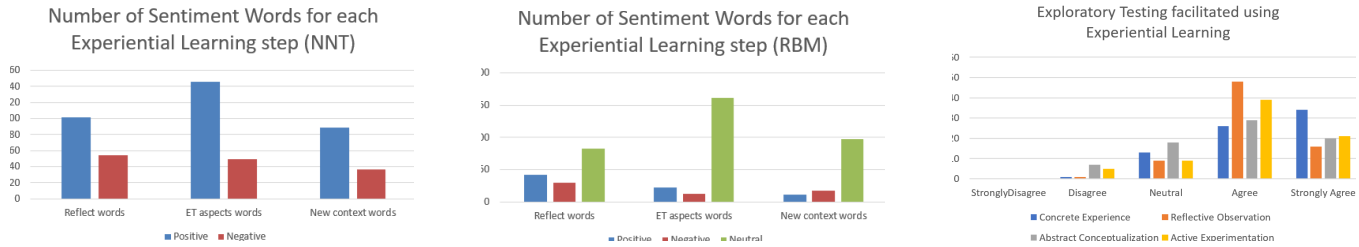
[1] https://pypi.org/project/SentimentAnalysis/

[2] https://huggingface.co/transformers/task_summary.html#sequence-classification



(a) (b) (c)

Figure 7: Experiential Learning results for one group. (a) Reflective; (b) Conceptualization; (c) Active experimentation.



(a) (b) (c)

Figure 8: Analysis results of seminars. (a) NNT analysis; (b) RBM analysis; (c) Experience learning steps and survey results.

4.2 Answers to the Research Questions

After conducting the investigation and obtaining the results, we may now provide our findings for the research questions.

Response for RQ1: Gamification proved itself like a viable strategy in terms of teaching. It offered the students conceptual progress bars for their studies and level of understanding. Being conducted at every level of learning (lecture by lego, laboratory by storytelling and metaphors, and seminars by dice game) the gamification allowed the integration of learning from various learning activities, students understanding the concepts.

Response for RQ2: Gamification offered the teachers a better quantification of the overall levels of interest and progress presented by the class. Engagement, sense of familiarity, and increased motivation have been obtained by integrating the student-friendly progress tracking mechanism via points and scoreboards. All hierarchy has been undergone in a productive competitive environment. Having a quantifiable and reachable goal, such as achieving as many points as possible, offers the students a palpable aim. This motivates them to be more responsive to the newly presented information in order to reach their goal. Having clear goals and destinations encourages the students to use the information gathering as means of achieving their ambitions.

By introducing elements of agile development and testing into the educational environment and so simulating an industry-like setting was received with increased interest, and students have mostly provided positive feedback.

Response for RQ3: The student's perception about using the game when facilitating learning by teachers in all three dimensions of learning, i.e. lectures, laboratories and seminars, were mostly positive. Students felt part of the team that needed to solve the tasks being given during the game, and bringing his/her contribution to solving the game. Collaborating or being competitive were both felt by the participants, very close to a real world setting. Entertaining and "happy with others ideas" were the other perspectives the students mentioned, thus realizing the importance of the team.

5 LESSONS LEARNED

The final aim of the investigation is to analyze and extract good practices with positive impact on the student's learning, and to see the challenges and improvements of the performed activities.

Lesson 1 - On the agile methodology

Laboratory activities included practices commonly used in agile development such as sprint planning and retrospective meetings. This was positively evaluated by students from two perspectives: (1) the tasks assigned were discussed and eventual problems examined within the team, (2) it provided an industry-like environment that can serve as experience for their future careers. It was especially useful in the current context of online education caused by the pandemic situation.

The students have highlighted that discussing the tasks in detail and debating or reflecting on issues has been effective and have helped them to turn in laboratory assignments. When reflecting on the various tools used in the laboratory activities mixed opinions were provided by students, some found this cumbersome while others considered it belonged to the experience of real scenarios of software development. The high number of students turning

in laboratory assignments and actively participating in the laboratory activities suggests that incorporating agile elements has been effective.

Lesson 2 - On the use of metaphors

The metaphors and the storification used during the laboratories helped students to better understand the software testing notions, being able to identify where the concepts are applied in a production environment and not only in a theoretical context. Thus, by the use of metaphors the meaning of the testing concepts were "discovered" by the students easily since the "static information" gained motion and context.

Lesson 3 - On the gamification facilitation

Lectures and seminar activities were designed and facilitated using various games oriented to learning software testing concepts. At the start of the activity the aim of the activity was stated, thus what was the game about and the main concepts to be learned, the students did a great job identifying ingredients, more specific notions regarding the main concept by reflecting on the way they interacted and asking themselves why they were doing some tasks, asking themselves what were the reasons behind their actions. Thus, they learned by discovery being guided by the teacher that is the facilitator of the learning. The students were involved in doing the tasks and wanted to play the game. One of the main results was that the lectures (that are seen as pure theoretical by majority of students) were more interactive and fun, and that the seminars (where it is often difficult to make the students talk) became more dynamic and the students were more engaged in the tasks.

6 CONCLUSIONS AND FUTURE WORK

The learning is a process of knowledge construction, students being actively involved. Engaging students in learning and staying in the flow acquire game-based learning and experiential learning approaches, making students reflect on their actions. The paper reports on the game-based activities used to facilitate learning various software testing concepts at all three levels: lectures, laboratories and seminars. The students' perception on games and storification, along with experiential learning approach certify that all those elements generate fun learning environments and attain the expected competences.

REFERENCES

[1] Mahmoud Abdulwahed and Zoltan K Nagy. 2009. Applying Kolb's Experiential Learning Cycle for Laboratory Education. *JEE* 98, 3 (2009), 283–294.

[2] Faten Alzaid. 2018. *The effects of gamification based formative assessment on motivation and vocabulary acquisition in ESL classroom.* Master's thesis. McGill University, Montreal.

[3] Scott W Ambler. 2007. Agile testing strategies. *Dr. Dobb's Journal, January* (2007).

[4] Thomas Auer and Michael Felderer. 2020. Towards a Learning Environment for Internet of Things Testing with LEGO® MINDSTORMS®. In *2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW).* 457–460. https://doi.org/10.1109/ICSTW50294.2020.00081

[5] Jonathan Bach. 2000. Session-based test management. (01 2000).

[6] Kesina Baral and Paul Ammann. 2020. Teaching a Testing Concept (JUnit) with Active Learning. In *2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW).* 411–411. https://doi.org/10.1109/ICSTW50294.2020.00073

[7] Jared R. Chapman and Peter J. Rich. 2018. Does educational gamification improve students' motivation? If so, which game elements work best? *Journal of Education for Business* 93, 7 (2018), 315–322.

[8] Shui Kau Chiu. 2019. Innovative experiential learning experience: Pedagogical adopting Kolb's learning cycle at higher education in Hong Kong. *Cogent Education* 6, 1 (2019), 1644720. https://doi.org/10.1080/2331186X.2019.1644720

[9] Alison Clear, Allen S. Parrish, John Impagliazzo, and Ming Zhang. 2019. Computing Curricula 2020: Introduction and Community Engagement. Association for Computing Machinery, New York, NY, USA.

[10] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. 2011. From game design elements to gamefulness: defining" gamification". In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments.* 9–15.

[11] Christo Dichev and Darina Dicheva. 2017. Gamifying education: what is known, what is believed and what remains uncertain: a critical review. *International Journal of Educational Technology in Higher Education* 14, 1 (20 Feb 2017), 9.

[12] Torgeir Dingsøyr, Sridhar Nerur, VenuGopal Balijepally, and Nils Brede Moe. 2012. A decade of agile methodologies: Towards explaining agile software development.

[13] Mayer Richard E. 2008. Applying the science of learning: evidence-based principles for the design of multimedia instruction. *Am Psychol.* 63, 8 (2008), 760–9.

[14] Martin Fowler, Jim Highsmith, et al. 2001. The agile manifesto. *Software Development* 9, 8 (2001), 28–35.

[15] Juho Hamari, Jonna Koivisto, and Harri Sarsa. 2014. Does gamification work?–a literature review of empirical studies on gamification. In *2014 47th Hawaii international conference on system sciences.* Ieee, 3025–3034.

[16] Elisabeth Hendrickson. 2013. *Explore It!: Reduce Risk and Increase Confidence with Exploratory Testing.* Pragmatic Bookshelf.

[17] Jean Heutte, Fabien Fenouillet, C Martin-Krumm, Ilona Boniwell, and Mihaly Csikszentmihalyi. 2016. Proposal for a conceptual evolution of the flow in education (EduFlow) model. In *The 8th European Conference on Positive Psychology (ECPP 2016).*

[18] Clayton Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 8.

[19] Cem Kaner, Jack L. Falk, and Hung Quoc Nguyen. 1999. *Testing Computer Software, Second Edition* (2nd ed.). John Wiley Sons, Inc., USA.

[20] A. Kolb and D. A. Kolb. 2013. *The Kolb Learning Style Inventory 4.0 Guide A Comprehensive Guide to the Theory, Psychometrics, Research on Validity and Educational Applications.* Experience Based Learning Systems.

[21] D.A. Kolb. 2014. *Experiential learning: experience as the source of learning and development.* Pearson FT Press, 2 edition, Upper Saddle River, New Jersey.

[22] Beata Lorincz, Bogdan Iudean, and Andreea Vescan. accessed June 2021. Experience Report on Teaching Testing through Gamification. = https://figshare.com/s/b958413e42020d879a59.

[23] Jenni Majuri, Jonna Koivisto, and Juho Hamari. 2018. Gamification of Education and Learning: A Review of Empirical Literature.. In *GamiFIN Conference 2018, Pori, Finland, May 21-23, 2018.* 11–19.

[24] Jenni Majuri, Jonna Koivisto, and Juho Hamari. 2018. Gamification of education and learning: A review of empirical literature. In *Proceedings of the 2nd international GamiFIN conference, GamiFIN 2018.* CEUR-WS.

[25] Mosima Anna Masethe, Hlaudi Daniel Masethe, and Solomon Adeyemi Odunaike. 2017. Scoping Review of Learning Theories in the 21st Century. In *Proceedings of the World Congress on Engineering and Computer Science.* 227–231.

[26] R. E. Mayer. 1998. Cognitive theory for education: What teachers need to know. *N. M. Lambert and B. L. McCombs (Eds.), How students learn: Reforming schools through learner-centered education* 84 (1998), 353–377.

[27] Richard E Mayer. 2002. Cognitive Theory and the Design of Multimedia Instruction: An Example of the Two-Way Street Between Cognition and Instruction. *New Directions for Teaching and Learning* 2002, 89 (2002), 55–71.

[28] Kathryn Nest. accessed 2020. The Dice Game. https://blog.gurock.com/teaching-software-testing-with-games/.

[29] C. Pahl. 2002. A Study Of Individual Learning Styles And Educational Multimedia Preferences An Experiment using Self-Directed Online Learning Resources.

[30] K. Palenikova, Z. Nasticka, Z. Vitezova, Smieskova E. Vallo, D., L. Rumanova, J. Zahorska, and L. Klimentova. 2017. Examples of good practices of experiential learning and student' opinions on experiential learning. In *Proceedings of the Applied Mathematics Conference, APLIMAT.* 1153–1163.

[31] Peter Schuh. 2004. *Integrating agile development in the real world.* Charles River Media, Inc.

[32] Sweden. accessed 2019. Mentimeter-Create interactive presentations, workshops, and meetings. https://www.mentimeter.com/.

[33] Erik Van Veenendaal, J Bach, V Basili, R Black, C Comey, T Dekkers, I Evans, P Gerard, T Gilb, L Hatton, D Hayman, R Hendriks, T Koomen, D Meyerhoff, M Pol, S Reid, H Schaefer, C Schotanus, J Seubers, F Shull, R Swinkels, R Teunissen, R van Vonderen, J Watkins, and M van der Zwan. 2002. *The testing practitioner.* UTN Publishers.

[34] Andreea Vescan and Camelia Serban. 2020. Facilitating Model Checking Learning through Experiential Learning. Association for Computing Machinery.

[35] Li-Fang Zhang, Robert J. Sternberg, and Stephen Rayner. 2011. *Handbook of Intellectual Styles: Preferences in Cognition, Learning, and Thinking.* Springer Publishing.