Gamification of business process modeling education: an experimental analysis

Gamification of Business Process Modeling Education: An Experimental Analysis

Giacomo Garaccione[1], Riccardo Coppola[1], Luca Ardito[1] and Marco Torchiano[1]

[1]Department of Control and Computer Engineering, Politecnico di Torino, Corso Duca degli Abruzzi, 24, Torino, 10129 , Italy.

Contributing authors: giacomo.garaccione@polito.it; riccardo.coppola@polito.it; luca.ardito@polito.it; marco.torchiano@polito.it;

Abstract

Gamification, the practice of using game elements in non-recreational contexts to increase user participation and interest, has been applied more and more throughout the years in software engineering. Business Process Modeling (BPM) is a skill considered fundamental for software engineers, with Business Process Modeling Notation (BPMN) being one of the most commonly used notations for this discipline. BPMN modeling is present in different curricula in specific Masters's Degree courses related to software engineering but is usually seen by students as an unappealing or uninteresting activity. Gamification could potentially solve this issue, though there have been no relevant attempts in research yet. This paper aims at collecting preliminary insights on how gamification affects students' motivation in performing BPMN modeling tasks and - as a consequence - their productivity and learning outcomes. A web application for modeling BPMN diagrams augmented with gamification mechanics such as feedback, rewards, progression, and penalization has been compared with a non-gamified version that provides more limited feedback in an experiment involving 200 students. The diagrams modeled by the students are collected and analyzed after the experiment. Students' opinions are gathered using a post-experiment questionnaire. Statistical analysis showed that gamification leads students to check more often for their solutions' correctness, increasing the semantic correctness of their diagrams, thus showing that it can improve students' modeling skills. The results, however, are mixed and require additional experiments in the future to fine-tune the tool for actual classroom use.

Keywords: Gamification, BPMN Modeling, Teaching, Information Systems

1 Introduction

Business Process Modeling is an essential skill for business analysts, as those who possess this skill can comprehend the various processes inside an organization and thus can make logical decisions and recognize areas that need improvements [1]. Proficiency in modeling business processes helps business analysts in defining effective models that can support software development by displaying the involved actors and the required process steps in a clear and understandable way. Business Process Model and Notation (BPMN) is a widely used notation for understanding and analyzing organizational processes. This standard uses flowcharting logic to graphically represent business processes, detailing the logic behind the procedure at both high and low levels. It is an effective tool for learning how to model and analyze organizational processes. Computer Engineering curricula in different universities include courses where BPMN modeling is taught. As other modeling activities in software engineering, Process Modeling is often perceived by students as unappealing and uninteresting if compared to coding activities [2]. At Politecnico di Torino, specifically, we have observed students struggling with the application of correct BPMN constructs, leading to low grades in the related tasks. We have thereby investigated the possible application of instruments to enhance students' motivation and – as a consequence – ability to learn. As reported by Jurgelaitis et al., Gamification can be considered a positive means of increasing motivation and students' grades in software modeling activities [3].

Gamification, defined by Deterding et al. [4] as *the use of game design elements in non-game contexts*, has been used with increasing frequency in both industrial and academic contexts as a way to increase user motivation, interest, and participation in activities that are generally seen as unappealing. The main goal of gamification is to increase involved users' productivity by creating a stimulating and engaging experience [5]. The usage of gamification-based approaches has shown some important advantages from the psychological user-experience perspectives in non-ludic activities, such as increased motivation, focus, and engagement, but also better performance and higher efficiency [6–11].

This article outlines a first attempt at using a gamified learning tool in an Information Systems course as a way to increase students' motivation and results in modeling activities. The tool implements commonly used gamification mechanics such as rewards, points, feedback, and progress indicators to guide students toward learning good modeling practices, with penalization also being used as an additional motivator. All the mechanics are supported by an evaluation engine that compares the solution submitted by a student with an expected representation and assigns points and penalties according to what it judges as modeled correctly and incorrectly. An experiment has been conducted with students of the course to assess whether gamification can lead to improvements in students' performances and modeling practices, as well as to assess how using a gamified tool is received by students.

This article has been conceived as the continuation of our previous article [12], in which we presented the prototype version of our tool, as well as a preliminary evaluation of the tool's usability and the effectiveness of gamification as a motivating strategy. The following changes have been introduced, concerning the original paper:

- The structure of the tool was changed from having three separate gamified versions of the tool with separate mechanics (progress indicators, leaderboards, unlockable rewards) to a single version that includes multiple mechanics (progress indicators, unlockable rewards, penalties, and feedback).
- The evaluation engine has been extended, with the existing rules being improved: the original version expected a specific number of elements of each type and connections between elements of specific types without checking more in detail, while the current version expands on these checks by expecting elements of a specific type, with a specific name, that belong to a specific pool and are also connected to other elements that have the same restrictions on type and naming.
- A non-gamified version of the tool has been implemented: this version offers a basic syntax evaluation of the produced diagrams as its only feedback mechanism; such an implementation is in line with other commonly used BPMN modeling tools such as Signavio Academic [13].
- A new evaluation experiment has been conducted using the current version of the tool and the non-gamified version: this evaluation involved 200 students compared to the 12 of the original article and focused on identifying whether gamification could improve students' modeling practices rather than on usability. The results of the experiment have been analyzed through statistical means.

The remainder of the paper is structured as follows: Section 2 provides a background explanation on gamification and gamified tools already used in education, Section 3 describes the gamified mechanics of the tool, as well as its evaluation engine that supports their implementation, while Section 4 describes the methodology applied during the experiment with which the tool was evaluated. Results of the experiment are presented in Section 5 and discussed in Section 6, while Section 7 provides the conclusions drawn from the experiment, as well as future plans regarding the tool.

2 Background

2.1 Gamification

Gamification involves the application of principles and mechanics typically found in games to enhance the user experience for non-recreational purposes.

The goal of incrementing the user's interest is achieved by addressing three of the basic human needs defined by self-determination theory: the need for competence, the need for autonomy, and the need for social relatedness [14, 15]. Addressing these needs resulted in the creation of specific game elements tailored to increase human motivation; we present below an explanation of the elements connected to each need:

- **Competence.** The experience of gaining mastery, feeling in control of the environment, and being effective in the activity one is performing. This need has been addressed by defining opportunities for players to earn points, awards, and badges; allowing comparisons among players through leaderboards is another example of how this need has been addressed [15];
- **Autonomy.** The feeling of being in control of one's own goals, rather than feeling compelled or controlled in one's behavior. Avatars, diverging story paths, and different ways to play have been created as a way to address this basic need [16].
- **Social Relatedness.** The need to have a close relationship and connection with other people. It has been addressed with meaningful storylines centered around the player and cooperative and competitive gameplay [15].

These game elements over the last decade have been applied to non-game contexts with increasing frequency.

There are many examples of frameworks for developing and assessing gamified systems: one of the most complete ones is the framework named Octalysis, theorized by Yu-Kai Chou [17], which defines eight main points, named **Core Drives** to define and evaluate a gamified system; these Core Drives are defined in Table 1.

2.2 Gamification for Software Engineering Education

Gamification has been increasingly used in the education sector in order to enhance the learning process [18]; benefits that stem from using gamification in a classroom include lowering the learning curve of complex topics [6] and increasing motivation in completing complex tasks [9]. Regarding Software Engineering, there have been many literature reviews and mappings performed in recent years focusing on the usage of gamification [19–24]. These reviews explore methods and strategies used in literature to identify common elements and effective solutions; moreover, they reveal that gamification is becoming more commonly used in Software Engineering, including the teaching of related disciplines.

Many frameworks for gamification of teaching have been proposed in recent years: an example of such a framework comes from Dubois and Tamburrelli [25], who describe an experiment using competition mechanics. The experiment focused on software quality and programming best practices through an integration with the Sonar [26]

Table 1 Core Drives defined in the Octalysis framework

Core Drive	Description
Epic Meaning and Calling	Users feel part of something greater and believe their actions contribute to a shared goal
Accomplishment	Users are driven by progress, learning, and achieving goals
Empowerment of Creativity	Users have the desire to express creativity, receive feedback, and have control over their actions
Ownership and Possession	Users feel ownership and control over something, leading to increased interest
Social Influence	Users desire to connect and interact with others, and feel a sense of belonging to a group
Scarcity and Impatience	Users are compelled to put more effort into obtaining things that are scarce or in high demand
Unpredictability	Users are motivated by novelty, surprise, and uncertainty
Loss and Avoidance	Users are motivated to act in a more careful way to avoid negative consequences for their actions

platform, allowing students to see scores connected to the quality of their code. Participants were divided into students who could only see their own scores and students who could see a leaderboard with all scores; students of the latter group obtained higher average metrics, leading authors to argue that competition is an effective addition to a gamified tool, as it can improve student performance.

A different framework, focusing more on theory topics, is OneUp by Dicheva et al. [27]: the platform can be used by instructors for defining exercises, with simpler exercises that can be attempted to train, and serious exercises with associated rewards. OneUp focuses on customization, as teachers can define either static exercises (true/false statements, multiple choice questions, matching) or complex, dynamic exercises whose correctness is checked by a program supplied by teachers. An experiment using the platform showed an increase in student motivation.

Uskov and Sekar [28] propose a framework for the gamification of a Software Engineering course: actions performed during the course (assignments, teamwork, deadlines, course grades, additional points for more challenging actions) are connected to the mechanics of gamification selected by the authors. These mechanics belong to three groups: progression (e.g. achievements, points, levels, trophies), feedback (e.g. challenges, bonuses, real-time feedback), and behavior (e.g. epic meaning, ownership, loss aversion, competition). The authors mention an experiment performed using the framework in a Software Engineering course: the positive results obtained in terms of student appreciation are encouraging, leading the authors to recommend using these mechanics as a way to enhance Computer Science courses.

Gasca-Hurtado et al. [29] describe the Pedagogic Instrument Design, a method that can be used to define teaching instruments that use experience-based learning and educational gamification to stimulate work and improve participation. The method works by defining five separate components named Preparation (learning goals, competencies that students will be expected to develop), Design (rules to achieve the game's goal, game materials and roles, steps for developing the game), Pilotage (pilot session for testing the game), Scheduling (improvements based on the pilot experience), and Assessment (evaluation of the results), with a step-by-step process that produces an instrument for each component. A case study where the PID was used in a session with software developers and software engineering students showed positive results in terms of enjoyment and creative thinking, as well as benefits in terms of learning the theory concepts presented.

Calderón et al. [30] present ProDec, a simulation-based serious game that aims to motivate students to learn the principles of software project management, following the ISO 21500 guidelines. The game offers a training environment where learners can familiarise themselves with the different stages of software development as well as environments for learning and designing game scenarios. The game covers five process groups of the ISO 21500 standard and has 100% coverage on processes that compose several subject groups of the aforementioned standard, while some of them are unfortunately not yet covered, as the authors report. Additionally, the authors mention plans of using ProDec in both industrial and university contexts to gather feedback for improvements, as well as to gauge any potential benefits it could bring in practice.

Maxim et al. [31] describe a serious game used to teach the basics of agile software development: this card-based game maps the sprints that are typically seen in scrum-based projects to virtual software development projects. The player faces an artificial intelligence opponent that reacts to the cards played with events that represent possible real-life obstacles (e.g. change requests, defects, accidents) to increase the challenge; such cards also become more common as the player becomes more experienced in order to increase motivation and interest. Additional game mechanics include rewards, badges, title changes, and experience points. The authors describe the implementation of this serious game and mention future plans to make it available as a downloadable Android application.

Lastly, the Framework for Gamified Programming Education or FGPE [32] provides the specifications for the gamification scheme and the exercise definition, a collection of gamified exercises covering different and popular programming languages, software for editing the exercises, and an interactive learning environment for the students. To the best of our knowledge, there is currently no example of usage of the FGPE in a classroom environment, as the only experiment focused on using the framework presents encouraging results in terms of its usability [33], with no evaluation in terms of learning benefits.

These various frameworks describe different and effective ways to implement gamification in a classroom environment, showing positive results in terms of student motivation, interests, and improvements in terms of grades and skills.

2.3 Gamification for Modeling Languages Education

The current research literature offers a few examples of gamified tools, as well as game-based approaches, for aiding teachers of modeling languages.

BPMN-Wheel, presented by Kutun et al. [34], does not fully fit the definition of a gamified tool, as it actually consists of a *serious game* (a game whose primary intended purpose is not enjoyment but learning [35], as opposed to gamification which entails the application of game-like elements to other established activities) used for teaching BPMN modeling. BPMN-Game consists of a competitive and cooperative board game that splits players into two teams: the teams take turns spinning a wheel to obtain theoretical questions on BPMN modeling which, if answered correctly, award them elements to build the target process or in-game currency. The team that is able to come closer to the intended process representation is the one that wins the game. An experiment conducted using the game showed that students, after playing the game, improved the quality of their modeled processes.

Other works that describe gamified tools for teaching modeling make use of the Papyrus platform [36], a modeling tool that supports many different languages; unfortunately, the BPMN standard is not supported by Papyrus, which covers only modeling languages derived from the Unified Modeling Language (UML).

Papygame, by Bucchiarone et al. [2], is an example of such a plugin for Papyrus whose games represent different modeling tasks; these games go on until they are either successfully completed or the student fails. Completing correctly a task awards players points and in-game progress, while failure displays a feedback screen that lists the different modeling errors, as a way to facilitate learning correct modeling practices. The tool employs a separate engine that implements the gamified mechanics, keeps track of players' progressions, and defines the rules of the different games; the engine allows for the definition of new games and exercises. The tool has been evaluated with a preliminary evaluation focused mainly on user experience, and the authors state that the results of the evaluation show promise for the future.

A second plugin has been proposed by Cosentino et al. [37], who developed a gamified tool that presents a challenge with different levels of increasing difficulty: these levels represent various topics connected to modeling concepts, and successful completion awards achievements and in-game rewards. Moreover, a noteworthy feature of the plugin is special attention to cheating prevention mechanisms that block the manipulation of user information; such a feature is something that is rarely seen in gamified tools. Regrettably, to our knowledge, there is no recorded usage of this plugin in a real educational setting.

An example of a serious game used to teach students good practices for conceptual modeling is Classutopia [38], a mobile application that implements a role-playing game as the context in which students are encouraged to learn. Students take part in a quest to protect the world of the game from an enemy in challenges that consist of correcting defective diagrams, with errors harming the player and effective corrections hindering the enemy; visual feedback is also present after evaluations, with correct changes and mistakes being colored green and red, respectively. An empirical evaluation of the game has been conducted with students of a Software Engineering course: such evaluation showed preliminary benefits, with the authors mentioning, however, that further work is still needed and that the sample used for the evaluation is too limited for the results to be taken as absolute.

ModelGame [39] is defined as a model that makes use of gamification to improve software modeling education, with a specific focus on the quality of the produced models. ModelGame defines challenges in the form of missions and scenarios that the student must overcome using software modeling, and uses points, progress indicators, and feedback as its gamified mechanics to improve the modeling process; furthermore, it defines 10 different quality metrics to assess the models such as scope quality (contextualization between the model and the challenge), syntactic quality (how much the model respects the modeling language's conventions), and semantic quality (how much the model matches with the domain of the challenge). A qualitative evaluation of ModelGame has been performed by having a survey among software modeling instructors, with 94.7% of the participants agreeing that using it in a classroom environment could be helpful by having gamification as a strategy for improving the quality of the models produced by students.

Marín [40] describes an example of a serious game used to teach students about the basics of BPMN modeling: students are tasked with completing BPMN models by selecting among a set of available diagram elements the ones that are most suitable to correctly model the process at hand.

The game offers a set of challenges of increasing difficulty, starting from a basic level, then going up to the medium level once the basic is completed, and to the advanced one in a similar fashion if the medium level is completed. Semantic questions that gauge the student's understanding of the available constructs are also present and contribute to the student's final score once the exercise is completed; leaderboards and achievements offer tangible rewards for the effort performed by students in the different exercises. After an evaluation performed by students of a software engineering course, the game was defined as easy to use, fun, and helpful for learning how to model BPMN diagrams; additionally, the presence of achievements and leaderboards was considered encouraging.

Lastly, BPMS-Game, by Mancebo et al. [41], is another tool that is worth mentioning: while it focuses more on the sustainability aspect of business process modeling rather than actual teaching, it still counts as an example of a tool that applies gamification to BPMN, using common staples of gamified tools such as badges, leaderboards, and achievements. BPMS-Game awards users that model business processes according to sustainability rules defined by game administrators with new awards and badges; unfortunately, however, it does not focus on correct syntax rules or good modeling practices.

This analysis of the current state of the art on gamified or game-based approaches to teaching modeling languages has shown that there is no example yet of a gamified platform for teaching BPMN modeling practices to students: the currently available tools either do not support the standard, define a different approach or are intended for a different audience and purpose.

2.4 BPMN Evaluation

Since our tool was designed for educational purposes a mechanism to evaluate the correctness of BPMN diagrams was required. We discuss an example of metrics that can be used for such a purpose, following the three main evaluation criteria defined by Dumas et al. [42]: syntactic quality, semantic quality, and pragmatic quality.

Syntactic Quality refers to how well a BPMN diagram respects the syntactic rules and guidelines defined by the standard of the modeling language (an example of which is defined by the Signavio association [43]). A process named *Verification* is performed to assess a diagram's syntactic quality: in this process, a diagram is evaluated according to behavioral correctness (possible sequences of process execution) and structural correctness (types of elements used and their interconnections) to ensure that every part of the process is modeled according to the standard rules.

Semantic Quality consists of the capacity of a BPMN diagram to accurately describe its associated domain. Although there are no established guidelines, it is possible to perform a *Validation* process, which compares the diagram to how the process the diagram is supposed to model is executed in the real world to determine whether the diagram is coherent with the real-world scenario and whether it includes all the relevant aspects of the real process.

Pragmatic Quality relates to the usability of a process. This usability is assessed through the *Certification* process, which gauges how easily the diagram can be understood, how easily it can be modified, and how accurately it represents the real-world process it is supposed to model. A diagram's pragmatic quality depends on details such as its graphical layout, its size, and its structural complexity; a diagram's Certification depends on how consistent its logical structure is with the graphical representation, as well as on how labels assigned to elements respect naming conventions (e.g. short labels, imperative verbs, business objects used as subjects).

3 Tool Description

The tool has been developed as a React-based [44] web application that uses the *bpmn.js* [45] JavaScript library to allow the creation of BPMN diagrams in a web page. The tool also employs an extension, *bpmn-js-bpmnlint* [46], that implements a linting functionality for evaluating the syntactic correctness of the diagrams modeled by students.

Figure 1 shows the different components that make up the tool: students interact with the client through their web browser and access an exercise page where they can model a diagram that aims to solve the exercise. Every evaluation is saved to the database by sending data to the connected server, and each interaction with the game mechanics is recorded as well.

Each evaluation also changes the way the exercise page is shown: feedback is given directly on the diagram modeled by the student, and errors change the data shown on the exercise menu; correct modeling choices affect positively the menu as well.

The interface of the tool that allows modeling diagrams, which is shown in Figure 2, is divided into a modeling canvas on which students can draw their diagrams using the different elements offered by the BPMN standard, and the tool menu, present on the left-hand side, which contains the gamified mechanics of the tool.

3.1 Gamified Mechanics

The tool implements four gamified mechanics: *Rewards*, *Penalty*, *Progress*, and *Feedback*.

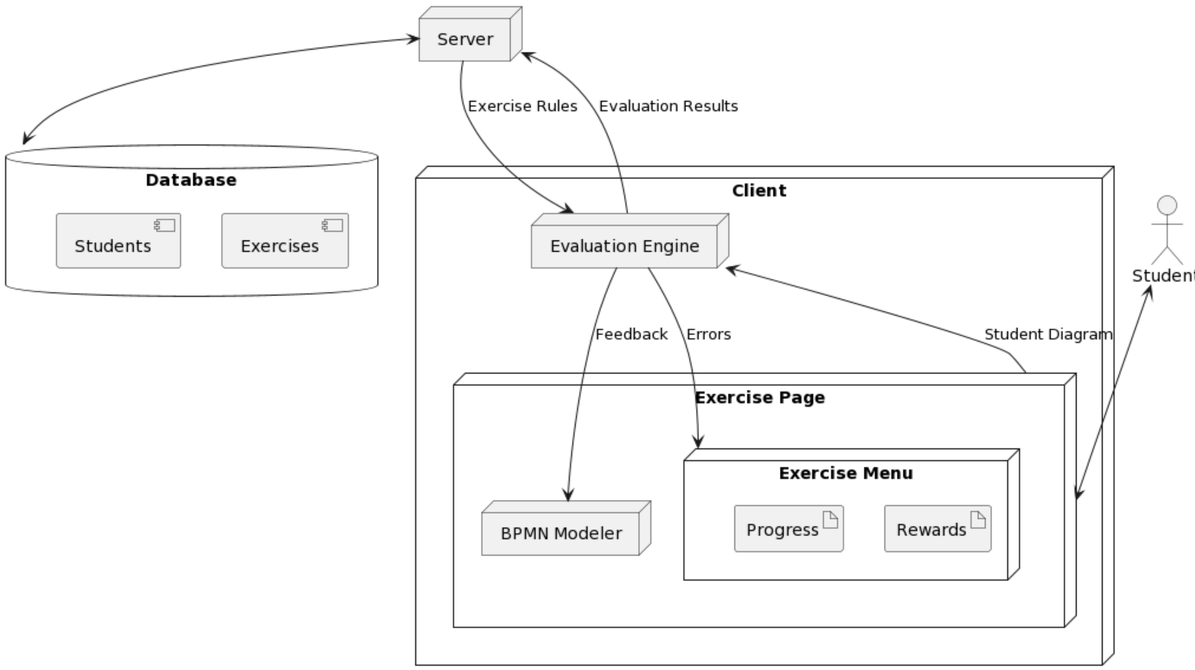


Fig. 1 Component diagram representing the tool

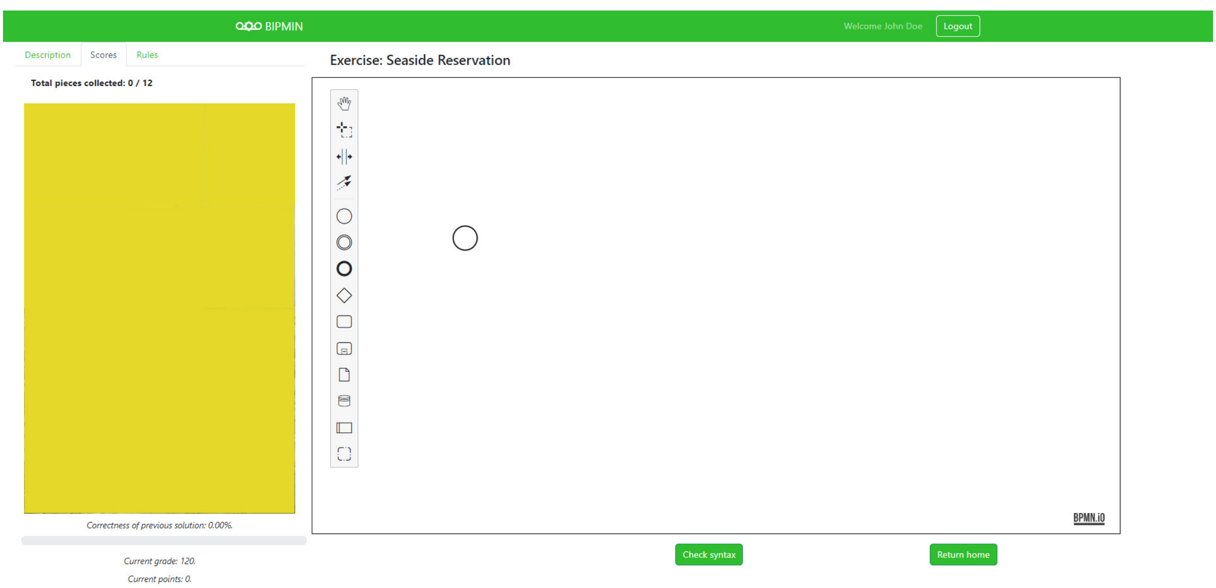


Fig. 2 Tool page for modeling diagrams

3.1.1 Rewards

The *Rewards* mechanic consists of points that are earned if a student models a specific *part* of the process in a way that is similar to the reference solution, the amount of points depends on how challenging the *part* is.

Once a student has obtained at least a point, they can spend the obtained points to purchase

Fig. 3 Scores section updated after purchasing pieces of the Jigsaw

pieces of a jigsaw shown in the tool menu, as shown in Figure 3. The amount of available points is specified by the *Current points* label and is updated any time the student either spends points or obtains new ones.

This mechanic is tied to the Core Drive named *Ownership and Possession* of the Octalysis framework. The concept of rewarding players for well-performed activities is quite common in gamified tools: an example of using rewards in a gamified tool comes from Ferreira and Oliveira [47], who describe an experiment using a tool for gamifying the teaching of exploratory testing. The tool's rewarding part involves both in-game and real-life rewards, as *treasures* are hidden in the classroom and can be found if students identify the necessary clues using the tool.

3.1.2 Penalty

The *Penalty* mechanic follows the same logic as the *Rewards* one, with the opposite behavior: the student is assigned a penalty for every *part* of the process that is modeled incorrectly. Said penalty is equal to the number of points that would have been awarded if the corresponding part had been modeled correctly, and the total sum of the penalties obtained for a single correctness check is subtracted from the student's *Current grade*: Figure 3 shows, in the *Current grade* label, a reduction in the grade after a correctness check where some *parts* of the process were not modeled in an expected way.

It is important to mention that, even though *points* and the *grade* follow the same logic (they both change after a correctness check depending on how many correct *parts* are present in the current diagram), they are separate from each other.

The grade getting reduced represents what would happen after an evaluation performed by a teacher, where errors would cause a reduction in the grade; the penalties are cumulative to act as a negative motivator whose goal is to deter students from aimlessly submitting the same solution over and over.

Points, instead, are a completely separate entity: their implementation is in line with common gamification strategies, in the sense that users are rewarded for performing correct actions. Points can be thought of as a sort of *currency* that is spent to purchase the pieces of the puzzle and obtain the final reward.

The adoption of this mechanic refers to the *Loss and Avoidance* Octalysis Core Drive, as tying a point loss to an incorrect performance leads to a higher effort spent in the modeling activity. Penalization mechanisms are not commonly used in gamified teaching tools: the aforementioned experience by Ferreira and Oliveira [47] is a rare usage of such a mechanic, with it not being a focal part of the gamified experience but an incentive to avoid reporting false positives.

3.1.3 Progress

The *Progress* mechanic consists of a progress bar, placed in the scores section of the student's User Interface, which displays the completeness percentage of the last solution checked by the student, allowing them to gauge what still needs to be improved, in combination with the *Feedback* mechanic.

The completeness is computed as the number of *parts* modeled correctly in the last submitted solution over all the *parts* that compose the process. An example of how the updated progress bar looks like can be seen in Figure 3.

This mechanic is tied to Octalysis' *Accomplishment* Core Drive, as it represents a visual indicator of a student's progression towards successful completion of a modeling assignment: increasing the progress towards success motivates students to try new modeling strategies.

An example of how to use progress indicators in a gamified tool is presented by Cacciotto et al. [48], who describe a framework for gamification of Graphical User Interface (GUI) testing. In their framework progress is used to show the coverage obtained by a tester in terms of elements of a page they have interacted with over the total amount of elements present in the page. Such a mechanism exploits a desire towards completion to motivate users of a gamified tool.

3.1.4 Feedback

Lastly, the *Feedback* mechanic can be considered as the most relevant gamified mechanic of the entire tool. Said mechanic is represented by a modal window which appears when a student performs the correctness check allowed by the tool: the modal window lists which *parts* of the process have been modeled correctly and which ones have either not been modeled or modeled in a way that the evaluation system does not register as

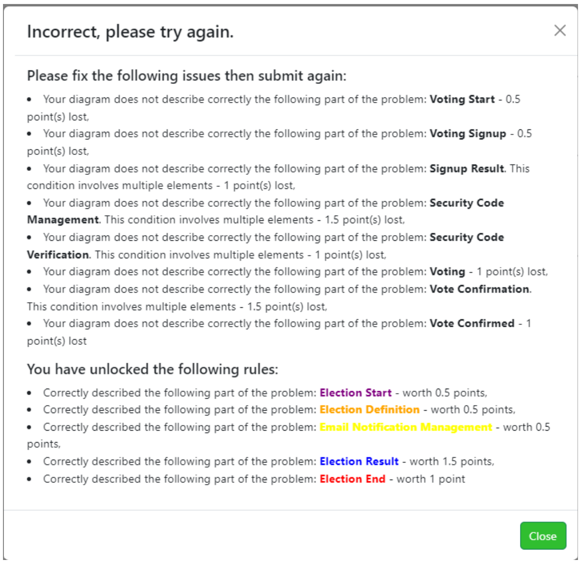


Fig. 4 Feedback modal showing *parts* of the process modeled correctly and incorrectly

correct; an example of how the modal looks like is displayed in Figure 4.

The rules that appear colored in the modal window also have their corresponding section of the diagram colored in the same way, clearly highlighting which element, or combination of elements, corresponds to which *part* of the process.

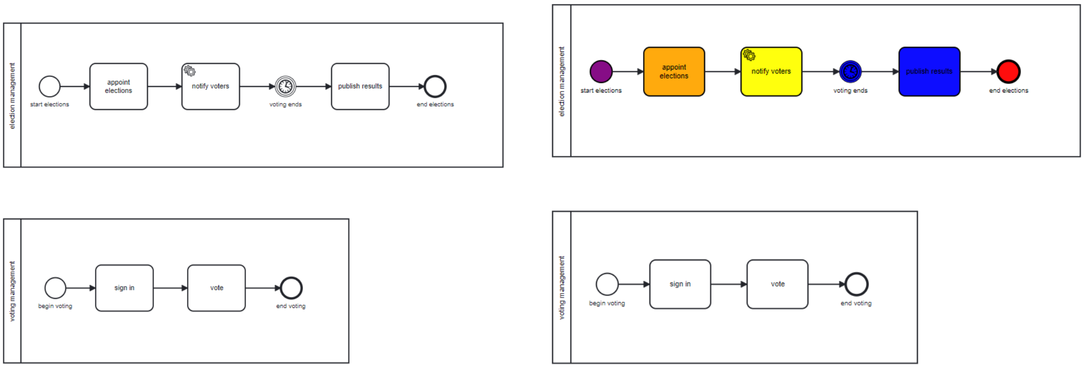
Figure 5(a) shows an example of a diagram submitted to a correctness check, while Figure 5(b) shows the same diagram with colored elements, indicating which *parts* of the process the student has modeled correctly; these *parts* also remain unlocked and have a dedicated section in the menu where the student can review them again. Each unlocked *part* is also highlighted with the same color as in the feedback modal and diagram, further reinforcing the mapping and allowing for an easier understanding of the process. Moreover, the feedback modal also lists the points associated with each *part*, directly connecting back to the *Rewards* and *Penalty* mechanics.

This mechanic is connected to the *Empowerment of Creativity* Core Drive, as the feedback given to students helps them understand if what they are doing is correct or not, allowing them to express their creativity. The Papygame tool [2] is an example of feedback being used as a beneficial addition to a gamified learning tool: a list of errors shown after each diagram evaluation is an effective way to improve students' modeling capabilities.

3.2 Evaluation Engine

The correctness check implemented in the gamified version of the tool makes use of an evaluation engine directly implemented in the tool that evaluates whether a diagram models correctly the various *parts* that make up the target process.

A *part* of the process is associated with either an element of a specific type or with a group of elements: for each element, there is a set of criteria that must all be satisfied to consider the corresponding part of the process modeled correctly. For example, the *part* of the process identified by *StartEvent_Reservation Start* is judged as modeled correctly if there is one element of type *StartEvent* that respects all the corresponding criteria.



(a) Diagram modeled by a student and submitted to a correctness check

(b) Updated diagram showing the correctly modeled parts in color, with wrong parts being left white

Fig. 5 Diagram modeled by a student before and after a correctness check

The various *parts* of the process are defined in a JSON object which lists all the criteria associated with each object, or group of objects, related to each *part* of the process, and the evaluation engine checks, for every rule, if the diagram contains one element whose type corresponds to the rule which also satisfies all the criteria; in case a rule is associated to multiple elements then the check is made on the simultaneous existence of all the defined elements that satisfy all the criteria.

The criteria used to evaluate elements are reported as follows:

1. **Label.** This criterion defines an array of strings and is satisfied if there is an element of the specific type that has a label that contains at least one of the defined strings.
2. **Parent.** This criteria defines an object that identifies another element of the diagram and that can have its own set of criteria to satisfy. This criterion is satisfied if there is at least one element that satisfies all the inner criteria defined and has an outgoing sequence flow towards the element corresponding to the rule.
3. **Target.** This criteria defines an object with its own set of criteria that must be satisfied. To satisfy this criterion, the diagram must contain at least one element that satisfies all the inner criteria and has an incoming sequence flow from the element corresponding to the rule.
4. **Element.** This criteria defines a string that specifies the type an element must have. This criterion appears in the list of inner criteria of elements that have **Parent** or **Target** as a criterion, specifying the type that the parent/target element must have.
5. **Pool.** This criterion defines an array of strings and is satisfied if there is an element of the type corresponding to the *part* of the process that belongs to a pool whose name contains at least one of the defined strings.

6. **EventDefinition.** This criterion is used only for elements that correspond to Events and defines a string that specifies which type (among, for example, Timer or Message) the Event must be for the criterion to be satisfied.
7. **MessageTarget.** This criteria defines an object that represents another element that must exist in the diagram, with its own set of criteria. The criterion is verified if there is at least one element that satisfies all the inner criteria and receives a message flow from the original element.
8. **MessageSender.** This criteria can be thought of as a combination of criteria **MessageTarget** and **Parent**, as it defines an object corresponding to an element with its own set of criteria to satisfy that must also have an outgoing message flow towards the element associated with the rule to evaluate.
9. **BoundingElement.** This criterion is used only by elements of type *BoundaryEvent*, and defines an object which specifies the element to which the event must be bound. There must be an element that satisfies all the inner criteria and also has a boundary event to satisfy this criterion.
10. **Interrupting.** This criterion is used only when referring to *BoundaryEvents*, and it defines a boolean value that specifies whether the event is a boundary-interrupting event or a non-interrupting one.
11. **BoundEvent.** This criterion defines an object that specifies the various criteria that a *BoundaryEvent* attached to the original event must satisfy to satisfy this criterion.
12. **GroupRules**. This criterion defines an array of objects where each object represents a group of elements, each with its own criteria to satisfy. It is used to define alternative options to model the same part of the problem, as well as to define elements that must be present together in the diagram.

Appendix A shows examples of how these criteria are expressed in practice.

As we mentioned back in Section 2.4, a possible way to evaluate BPMN diagrams is to focus on their Syntactic, Semantic, and Pragmatic Quality.

The evaluation engine assesses both the Syntactic and Semantic Quality of a diagram, with the latter being evaluated with rules following the set of criteria we described in this Section.

This evaluation mechanism has some limitations, as unpredictable factors such as human errors in naming elements make defining all possible options for labels a complex activity. Moreover, the same process, or part of a process, may be modeled in different equivalent ways and it is not feasible to cover all combinations of elements and names.

Pragmatic Quality, instead, is evaluated more effectively through human inspection, making it

Table 2 GQM template for the experiment

Object of Study	Learning of BPMN Modeling
Purpose	Compare the traditional and gamified approach
Focus	Productivity, Correctness
Context	Modeling tools
Perspective	Teachers

something that cannot be evaluated in an automated way as easily as the other two metrics. The engine considers only the contents of a diagram during its evaluation, paying no attention to the actual positioning of elements, size, or naming conventions, as the presence of specific words is judged more important.

4 Experiment Design

The purpose of the experiment is to investigate the impact of gamification on increasing motivation in learning BPMN modeling and on productivity and correctness as a consequence. We have defined our research objectives using the Goal-Question-Metric template [49] as reported in Table 2. The goal of the experiment is to assess whether gamification affects the students' productivity in producing BPMN diagrams and the correctness of those diagrams; in particular, if gamification positively impacts these two metrics. The experiment results are interpreted from the point of view of teachers of BPMN modeling and related topics.

4.1 Research Questions

We define the following research questions to frame the experiment design:

- *RQ1:* Does gamification improve the students' *productivity* in modeling BPMN diagrams compared to non-gamified BPMN modeling?
- *RQ2:* Does gamification improve the *correctness* of the diagrams modeled by the students compared to non-gamified BPMN modeling?
- *RQ3:* Is the gamified experience *well perceived* compared to non-gamified BPMN modeling?
- *RQ4:* Does gamification improve *enjoyment* and *perceived usefulness* compared to non-gamified BPMN modeling?

4.2 Design

The study focuses primarily on the BPMN diagrams modeled by the students. The investigation needed to perform the study was organized as a within-subjects 2x2 full factorial (crossover) experiment where the treatment was administered to participants through two versions of a tool, without gamification (*Vanilla*) and with it (*Gamified*).

The *Vanilla* version of the tool offered basic feedback, which reported only the syntactic errors rather than both syntax and semantic errors: this was decided, rather than have limited feedback related to both error types so that the *Vanilla* version was closer to the modeling tool regularly used by students during the course activities.

The tool, called Signavio Academic [13], offers modeling capabilities for different diagrams (BPMN, UML class diagrams, activity diagrams),

Table 3 Experiment Design

	Task 1		Task 2	
	Exercise	Tool	Exercise	Tool
Group 1	Seaside	Vanilla	Elections	Gamified
Group 2	Seaside	Gamified	Elections	Vanilla
Group 3	Elections	Vanilla	Seaside	Gamified
Group 4	Elections	Gamified	Seaside	Vanilla

and allows users of its BPMN modeler to check whether their diagrams are following basic BPMN syntax rules. It was for this reason that we decided to compare a version similar to the one students were used to (a BPMN modeler that offers only syntax evaluations) with a gamified version that offered more detailed feedback and additional mechanics.

All the participants received both treatments and had to perform both exercises in two consecutive tasks (periods), resulting in two possible sequences: Gamified-Vanilla and Vanilla-Gamified. In addition, there are two possible orders of exercises (objects) for a total of four different groups. The experimental design is summarized in Table 3. The participants were assigned randomly to the four groups.

After the second exercise session, the participants were asked to answer a questionnaire about their experience, including the GAMEX questionnaire [50] and other questions regarding the different gamified mechanics. The objective of the questionnaire was to investigate the perception of the gamified experience, as well as the influence the various mechanics had on the experience itself.

4.3 Operationalization of Variables

Our experimental design is based on three independent variables. One main factor (*Treatment*) corresponds to the testing tool and its two possible uses: with or without gamification (Gamified/Vanilla). Two controlled co-factors related respectively to the *Exercises* used (see section 4.5) and the *Order* in which the tasks are performed (Vanilla-Gamified or Gamified-Vanilla). The dependent variables evaluated in the experiment relate to the proposed gamified approach's eventual benefits to productivity and correctness. Additionally, the GAMEX aspects are used to assess the students' perception of the gamified experience, while the final set of questions, including open-ended ones, gauges the enjoyment and perceived usefulness of the gamified tool. Since students produced many versions of diagrams for each exercise, we used the final diagram produced by each student for each of the two exercises for the analysis.

Productivity was measured in the experiment using two variables: the *Size* of a diagram, computed as the count of all the elements included in the diagram itself, and the *Number of submissions* made for each exercise; submissions were divided in standard syntax checks, available in both versions, and correctness checks available only in the gamified version of the tool, and only possible in case the diagram passed the syntax check.

The solutions we defined for the two exercises were planned in such a way that there would be a necessary minimum level of diagram elements for a diagram to be considered correct; this ensures that there cannot be excessive variability in terms of the expected number of elements present in a diagram.

Furthermore, the solutions were structured so that there was no variance in terms of different combinations and flows of elements being allowed for the same process part (e.g. using two tasks to model two activities was the only allowed option, with having a single task that encompasses the two activities not being considered correct); the only variance allowed was in the names given to the different elements, in the sense that using synonyms was not counted as a mistake.

Thus, by having an almost fixed expected number of elements for a solution to be considered correct, we can assert that having more tasks equates to being more productive, and by allowing variations only in the naming and not in the different types of elements, we reduce the risk of marking as incorrect a diagram that is actually correct, because a correct diagram needs to closely match the expected solution.

In the experiment, we defined *Correctness* as the combination of two variables: *Syntactic correctness* and *Semantic correctness*. The former has been measured as the number of syntax rules that have been respected by a diagram over the total amount of syntax rules. The syntax rules we have defined for the evaluation are divided into two groups: rules that are always checked and rules that are only checked if the diagram contains an element of a specific type; these rules are listed in Tables B1, B2, and B3 of Appendix B.

Regarding *Semantic correctness*, instead, we defined a separate set of rules corresponding to a specific part of the process represented by each exercise and computed the correctness as the number of rules that were respected (that is, modeled in a way that the correctness evaluation recognized as correct) over the total number of rules. Each exercise has a specific set of rules, which we present in Tables B4 and B5 of Appendix B.

To answer RQ3 and to assess how students perceived the gamified experience we made use of the GAMEX questionnaire. The questionnaire provides questions that evaluate six different aspects of gamified experiences: Enjoyment, Absorption, Creative Thinking, Activation, Possible Negative Effects, and Dominance; answers were collected in the form of Likert-scale values ranging from 1 (*Strongly Disagree*) to 5 (*Strongly Agree*). The full list of questions is visible in Tables C6, C7, and C8 of Appendix C.

To answer RQ4 we included in the questionnaire a set of questions centered around the four gamified mechanics present in the tool (rewards, penalty, progress, and feedback): for each mechanic, the questionnaire included three questions, asking for the student's opinion regarding the specific mechanic's influence on the experience, its perceived usefulness, and how much the student had appreciated the presence of the mechanic.

Moreover, the questionnaire also included two open questions: one that asked participants to report any issues they had during the experience, and one where participants were encouraged to leave any comments or opinions that they felt could be useful for improving the tool and the gamified experience.

Appendix C lists the questions related to the gamified mechanics in Table C9: those questions allowed answers in the form of Likert-scale values, using the same range used for RQ3. The two open questions are also included in the same Appendix.

4.4 Participants and Sampling

We recruited participants for the experiment via convenience sampling. All the participants were students enrolled in the master's level in management engineering *Information Systems* course at Politecnico di Torino. The students were encouraged to participate in the experimentation by assuring them 2 additional points (out of 30) for the final grade of the exam. Participants were a total of 200 students, with 105 male students and 95 female students. We clarified to the students that participation in the experiment was optional since the very beginning of the course and that there would not have been any negative consequence for skipping the experiment. Therefore, we can reasonably assert that the students not partaking in the experiments did not perceive any negative feeling such as being left out or *FOMO* (Fear of Missing Out).

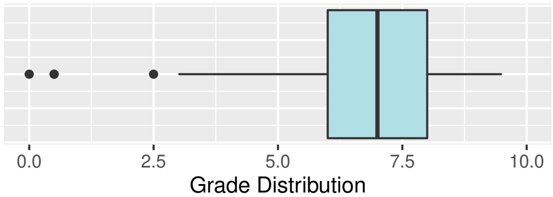


Fig. 6 Distribution of points for BPMN tasks in exams

We also analyzed the participants' skills in BPMN modeling by observing the sum of the grade assigned to BPMN-related exercises in the course's written exam (up to eight points) with the grade assigned to the BPMN exercise of an optional course assignment (up to two points). Results, which are reported in Figure 6, show that most students are fairly skilled in BPMN modeling, with the mean value being 7 points out of 10 total.

4.5 Experimental subject exercises

The two exercises selected for the experiment consist of two descriptions of real-life situations that made use of an Information System; the two exercises required students to model a BPMN diagram following the rules and conventions they were taught during the course.

The two exercises were selected among the ones defined by the course's supervisors, since their difficulty was judged to be average, allowing for the average student to be able to solve them in an adequate way.

The full text of the two exercises is reported in Appendix D, and examples of solutions accepted by the evaluation engine are available as an online resource[1].

4.6 Instrumentation

For the experiment, we developed a React-based web application, of which we provide a replication package as a Docker container available online[2]; additionally, we provide the data analyzed in the experiment (diagrams produced by students, results of the evaluations, answers to the questionnaire) as an online resource[3]. The experiment was conducted in a mixed way (both in-presence at our university's computer laboratories and remotely for students who could not attend), with students accessing the website via either the laboratories' devices or their own.

Data was collected by the tool and fetched at the end of the experiment in order to be analyzed; the collected data consisted of the various diagrams modeled by the students for each experiment and submitted to the tool for analysis.

[1] https://doi.org/10.6084/m9.figshare.22811123.v3
[2] https://hub.docker.com/r/giacomogaraccione/bipmin
[3] https://doi.org/10.6084/m9.figshare.24893601.v2

Table 4 Null hypotheses for the experiment

Name	Hypothesis
H_{s_0}	Gamification has no statistically significant impact on the *Size* of the diagrams submitted by students
H_{as_0}	Gamification has no statistically significant impact on the *Number of Submissions* made by students
H_{syc_0}	Gamification has no statistically significant impact on the *Syntactic Correctness* of the diagrams submitted by students
H_{sec_0}	Gamification has no statistically significant impact on the *Semantic Correctness* of the diagrams submitted by students

Additional details about the tool's implementation have been described in our previous work [12].

4.7 Hypotheses

To answer research questions RQ1 and RQ2, we have formulated the hypotheses reported in Table 4, for which we refer back to the variables defined in Section 4.3. Quantitative metrics are used to answer these questions through the use of formal statistical analysis, discussed further in Section 4.8.

Regarding research questions RQ3 and RQ4, we did not formulate any hypotheses, so the discussion of the results regarding those questions will focus on the distribution of answers given to the questionnaire, as there is no quantitative process to execute to assess the results.

4.8 Analysis Method

We adopt a non-parametric approach to test the hypotheses since we expect most variables to be not normally distributed. All statistical analysis is performed using the statistics tool, R [51].

Following the recommendations given by Vegas et al. [52] for analyzing the results of experiments whose design is a full factorial crossover, we analyzed the data using a repeated measures linear mixed model also considering the exercise and order (vanilla-gamified or gamified-vanilla) design factors to deal with the possible threats to validity deriving from the design.

Regarding RQ3 and RQ4, which were evaluated using a questionnaire with answers that are not easily analyzable using statistical hypothesis testing, we opted instead for descriptive statistics, reporting the results using stacked diverging bar charts [53] for the Likert-scale questions. Moreover, since RQ4 involves open-ended questions, we performed open coding [54] on the answers given by students, analyzing the content of each answer and identifying themes based on repeated words, common meanings, and contextual information.

4.9 Threats to Validity

We discuss here the potential threats to the study's validity according to the four categories defined by Wohlin et al. [55].

Threats to Internal Validity concern internal factors that may affect a dependent variable that the study did not consider. The experiment's crossover design can lead to threats connected to fatigue and learning effects on the participants.

The experiment was conducted both on-site and remotely, to meet the availability of the participants, and tasks had to be executed consecutively, each with its own time limit. We can thus expect the presence of fatigue effects on the participants. The presence of such bias can be detected by looking at the effects of the order of tasks in the analysis; the fact that the two tasks were executed immediately one after the other also means that there might be a carryover effect (e.g. a treatment being administered before the effects of a different treatment administered previously have been completely received).

The construction of the experiment design (2X2 factorial design) mitigates learning biases; however, the fact that the two tasks were performed with tools that share a common base (the modeling platform and the syntax check) means that a small learning effect on the results of the second task is to be expected, due to the familiarity with the mechanics implemented in the base tool. The tasks performed last might prove to be more (or less) effective than those performed first due to learning and fatigue effects [52]. The presence of such an effect might be revealed by an analysis of the specific task execution order (vanilla-gamified or gamified-vanilla).

Students who took part in the experiment were awarded two additional points on their final course grade: we do not consider the fact that this promise would have attracted more motivated students a threat towards the results of the study [56].

Another possible threat to the validity of the experiment comes from the two exercises we chose: there is no guarantee that exercises of higher, or lower, difficulty would yield comparable results to the ones we obtained. Moreover, the selected exercises had a comparable level of difficulty, but the analysis of the results might show that one was, in practice, easier than the other.

After the questionnaire results were obtained, we performed an analysis of face validity and content validity of the various questions where each author expressed a rating for each question: for face validity, authors expressed whether each question was relevant or not, and for content validity authors had to choose whether questions were necessary, useful but not necessary, and not necessary. The analysis showed that questions Q2.3, Q2.4, Q5.2, and Q6.2 of the GAMEX questionnaire were unanimously considered to not be relevant and that questions Q2.3, Q2.4, and Q5.2 were unanimously judged to not be necessary; those questions were thus excluded when computing the distribution of answers to the GAMEX questionnaire. The content of each removed item can be seen in Appendix C at the end of this paper.

Threats to External Validity concern whether the results of the study can be generalized rather than be applicable only to the specific sample of participants involved.

Moreover, the results we obtained are in part influenced by the sample of participants in the experiment: the tool could have different results if it were to be used by students with different skill levels in BPMN modeling.

Lastly, we cannot assume that the benefits brought by the tool can be generalized to contexts different from a university course: it is not possible to say whether the tool would be as effective in real-world modeling as it was for classroom exercises.

Threats to Construct Validity concern the extent to which the measures selected for the study actually represent the observed construct. There is no guarantee that the metrics selected to answer research questions RQ1 and RQ2 are the most effective, as other metrics could prove more suitable.

Regarding *Productivity*, we could have also considered the time spent for each exercise: each exercise had a maximum time allocated of 45 minutes, so we could have considered whether students were able to complete the exercises in a lower amount of time. However, time spent does not necessarily correspond to a correct diagram, so we did not include this measure in the analysis.

Regarding *Correctness* metrics, there is no guarantee that the rules we selected were the most effective for measuring a diagram's correctness. For example, semantic correctness was measured by looking for elements satisfying properties such as the name, specific words, or being connected to other elements. There could have been different ways to measure this correctness, such as looking for elements modeled in a way deemed wrong. Moreover, the set of accepted modeling options is not exhaustive, as it does not cover every possible correct alternative way to model a specific part of one of the processes under test.

Threats to Conclusion Validity concern the ability to conclude from the results of the study. We employed nonparametric statistical tests that have essentially no statistical prerequisite. All measures were collected automatically, meaning that we expect no human errors to impact their collection.

Since the experiment was designed around two different operational environments, the participants were randomly assigned to the four groups, with no statistical difference among the groups regarding BPMN modeling expertise. The random allocation is a confounding factor that could have influenced the statistical results, as a different allocation could have led to different statistical test results. However, the number of participants and their skill level distribution in each group leads us to consider this threat improbable but not one that can be ignored entirely.

Table 5 Summary statistics for *Productivity* metrics

		Seaside		Elections		All	
		V	G	V	G	V	G
Size	Mean	61.7	61.2	48.6	50.5	55.1	55.9
	Median	60.0	60.0	48.5	50.5	54.0	56.0
	Std. dev	13.3	12.3	10.6	10.6	13.7	12.7
Submissions	Mean	6.5	14.6	5.7	14.5	6.1	14.5
	Median	6.0	11.5	5.0	12.0	5.0	12.0
	Std. dev	4.3	10.9	3.4	10.1	3.9	10.5

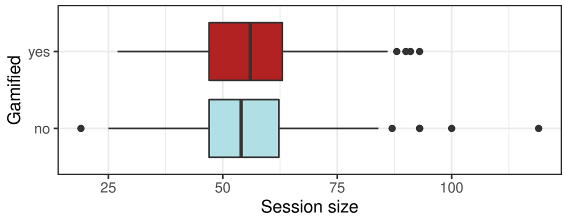
5 Results

5.1 RQ1 - Productivity

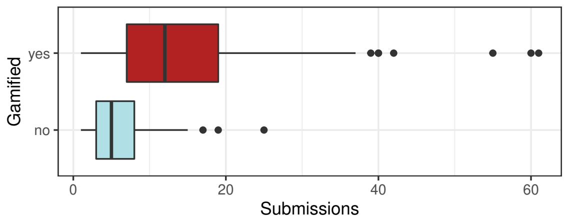
In Table 5 we report the mean, median, and standard deviation for the metrics selected to answer RQ1, e.g. the size of the final diagram submitted by students for each exercise and the number of different checks made in each exercise (syntax check and gamified correctness check).

Figure 7 shows box plots for the distribution of each metric aggregated by treatment.

From observing the two box plots, it can be deduced that gamification has an impact on the number of submissions made by the students, while the same cannot be said for the *Size* metric, as the two graphs appear almost identical. These observations are confirmed by the mean values of the two metrics: the mean diagram size increases in a negligible way when using the gamified version of the tool (55.1% to 55.9%), while the number



(a) Box plot for the *Size* metric



(b) Box plot for the *Submissions* metric

Fig. 7 Box plots for RQ1 metrics

Table 6 Results of ANOVA for RQ1 metrics: Size and number of Submissions

Factor / vars:	Size (H_{s_0}) coeff.	p-value	Submissions (H_{as_0}) coeff.	p-value
Treatment	-0.3775	0.2170	-4.215	< **2e-16**
Exercise	-5.9700	< **2e-16**	-0.2100	0.2525
Order	0.4750	0.4230	-2.2300	**0.0056**

of submissions increases from 6.1 to 14.5 (137% increase).

The results of the Anova test for the variables related to RQ1 are reported in table 6.

We observe a statistically significant effect of the application of gamification on the number of submissions (checks operated by the students) per exercise; thus we reject the null hypothesis H_{as_0} ($p < 2e-16$). No significant effect could be detected on the size-dependent variable; thus we cannot reject H_{s_0} ($p < 0.21$).

For what concerns confounding factors, we observe that the performed exercise had a significant effect on the size of the provided solutions ($p < 2e-16$): we can deduce that one of the two selected exercises required more diagram elements to be modeled correctly, and thus that it had a higher level of complexity compared to the other exercise.

Moreover, we observe that the order had a significant effect on the number of submissions ($p = 0.0056$): we can interpret this result as students being more motivated in checking for the correctness of their diagrams when moving from the base version of the tool to the one with gamification enabled.

Answer to RQ1: The statistical analysis shows that gamification has an impact on the number of diagram evaluations performed by the students. Gamification has no impact on the size of the diagrams produced, which is something that depends on the exercise.

5.2 RQ2 - Correctness

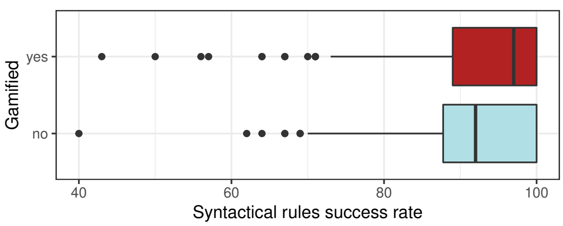
In Table 7 we report the mean, median, and standard deviation for the metrics selected to answer RQ2, e.g. the syntactic and the semantic correctness of each final diagram submitted by students during each exercise. Figure 8 shows box plots for the distribution of each metric aggregated by treatment.

Table 7 Summary statistics for *Correctness* metrics

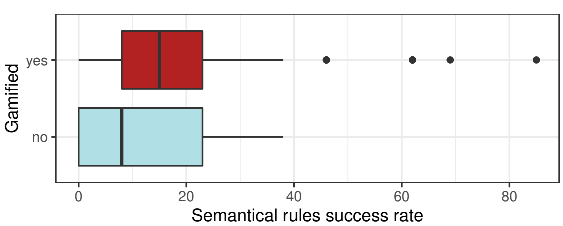
		Seaside		Elections		All	
		V	G	V	G	V	G
Syntactical (%)	Mean	90.8	91.7	91.4	92.6	91.1	92.2
	Median	92.0	93.5	92.0	100.0	92.0	97.0
	Std. dev	11.0	10.5	9.01	10.9	10.0	10.7
Semantic (%)	Mean	13.2	19.4	11.2	13.0	12.2	16.2
	Median	15.0	15.0	8.0	15.0	8.0	15.0
	Std. dev	11.8	17.2	8.8	10.4	10.5	14.6

Table 8 Results of ANOVA for RQ2 metrics: Syntactical and Semantic Correctness

Factor / vars:	Syntactical (H_{syc_0}) coeff.	p-value	Semantic (H_{sec_0}) coeff.	p-value
Treatment	-0.5250	0.2480	-2.0170	**0.0211**
Exercise	0.3650	1.0000	-2.0980	< **2e-16**
Order	-1.7100	0.9800	-1.0250	0.3180



(a) Boxplot for the *Syntactical Correctness* metric



(b) Boxplot for the *Semantic Correctness* metric

Fig. 8 Boxplots for RQ2 metrics

As can be observed, gamification has an impact on the *semantic correctness* of the students' diagrams, while the same cannot be said for the *syntactical correctness*, which sees a negligible increase in mean value compared to the former (91.1% to 92.2% and 12.2% to 16.2%).

We also observe an interesting variance between the mean values of syntactical and semantical correctness for each exercise as well as in general: this result stems from the fact that semantic correctness depends on a student's understanding of BPMN and the experiment was performed at the end of the course, meaning that students were more likely to know the necessary concepts to avoid syntax errors.

Semantic errors, on the other hand, are caused by missing elements that match those of the expected solution, meaning that students need to model a diagram that is close enough to the expected one to have high semantic correctness; these lower values show that having only one reference solution is currently a strong limitation of the tool.

The results of the ANOVA test for the variables related to RQ2 are reported in table 8.

We observe a statistically significant effect of the application of gamification on the semantic correctness of the submitted solutions; thus we reject the null hypothesis H_sec0 ($p = 0.0211$). No significant effect could be detected on the syntactical correctness of the provided solutions ($p = 0.2480$).

For what concerns confounding factors, we observe that the performed exercise had a significant effect on the semantic correctness of the provided solutions ($p < 2e - 16$): thus, we can confirm that one exercise had a higher level of complexity compared to the other, as the higher correctness rate can be seen as a consequence of one exercise being easier to model correctly. The absence of significant effects on syntactical correctness means that such a metric does not depend on the chosen exercise but rather on the student's capabilities.

We observe that the order had no significant effect on either syntactical or semantic correctness: we can deduce that changing from a non-gamified version of the tool to a gamified one or vice-versa does not impact students' performances in a significant way, in comparison to using the gamified version in place of the basic one.

Answer to RQ2: The statistical analysis identified an increase in semantic correctness of the diagrams when using the gamified version of the tool in comparison to the vanilla version. This metric has also been influenced by the exercise, while syntactic correctness had no statistically significant changes depending on any of the variables.

5.3 RQ3 - Overall Perception of the Gamified Experience

Regarding research questions RQ3 and RQ4, the total amount of answers collected amounts to 199, as one participant did not answer the questionnaire.

Figure 9 shows the distribution of the answers to the GAMEX questionnaire: to compute the distribution we have considered, for each participant, the rounded mean of the answers given to each question for each aspect of the questionnaire (e.g. for the *Enjoyment* aspect, we computed the rounded mean of scores given to questions from Q1.1 to Q1.6).

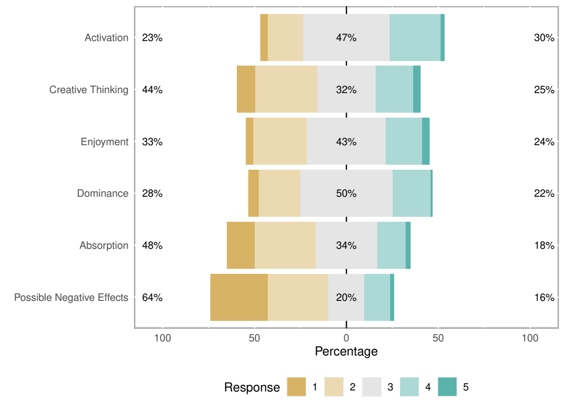


Fig. 9 Distribution of the answers to the GAMEX questionnaire

At a glance, results do not appear to be exceedingly positive, seeing as four groups out of six have high *Neutral* average scores (the *Dominance* group particularly stands out, with a total of 50% for the *Neutral* answer)

Distributions of positive answers are also lower than those related to negative answers for some groups: a glaring example of this is given by the *Absorption* group, with as much as 48% (15% *Strongly Disagree* + 33% *Disagree*) of students giving an average negative rating to said question group.

This result may be linked to the fact that questions try to gauge how much participants were immersed in the gamified experience: it is reasonable to assume that a laboratory experience may not be exactly the most immersive environment, and thus participants may not have felt particularly immersed in the tool.

A positive result, however, is the one achieved by the *Possible Negative Effects* group: only a total of 16% (14% *Agree* + 2% *Strongly Agree*) of students have answered with an average score that indicates a negative or frustrating experience, possibly implying that the gamified experience, while not exactly immersive, can be thought of as not frustrating, at the very least, which is a necessity for a gamified tool that aims to make unappealing activities more enjoyable.

Answer to RQ3: The answers to the GAMEX questionnaire show that students did not perceive the gamified experience as a particularly engaging one. The most encouraging result is a lack of negative effects, while other GAMEX aspects show a distribution of answers more on the neutral side.

5.4 RQ4 - Enjoyment and Perceived Usefulness

The distribution of answers to the questions related to gamified mechanics can be seen in Figure 10, which reports the distribution of answers to the questionnaire; as a way to represent the results in a more compact format, as well as to facilitate discussion, we considered, for each participant, the rounded mean of the answers

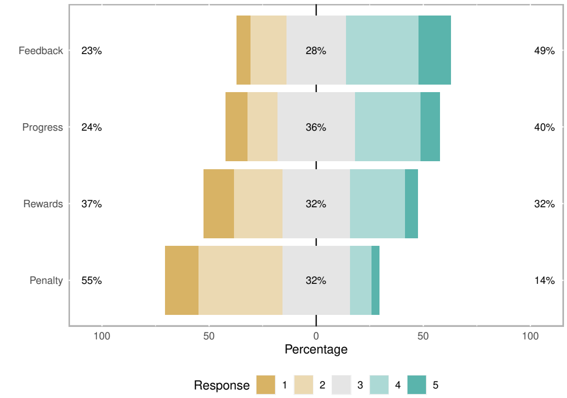


Fig. 10 Distribution of the mean answers related to the gamified mechanics' three constructs (influence, perceived usefulness, appreciation)

for each gamified mechanic and then plotted the distribution of those mean values.

We observe that the mechanics have achieved different results, as two mechanics appear to have been appreciated more, one mechanic shows a distribution of answers more on the neutral side, and another mechanic seems to have been mostly disliked.

The *Feedback* and *Progress* mechanics are those that have obtained positive results, with the former being appreciated by 49% (34% agreeing and 15% strongly agreeing) of the participants and the latter by 43% (9% strongly agreeing and 31% agreeing) of them.

The distribution of answers regarding the *Rewards* mechanic, instead, presents results that are more on the balanced side: almost a third of the participants answered, on average, positively to questions concerning rewards, while slightly more than a third (23% disagreeing and

14% strongly disagreeing) of the students did not appreciate the mechanic, resulting in an even split of answers between positive, negative and neutral.

Lastly, the *Penalty* mechanic stands out with an overwhelmingly negative distribution of answers: more than half of the students answered negatively to the questions connected to penalization, with 39% of them disagreeing and 16% strongly disagreeing with the questions; with as little as 14% of participants showing appreciation for the mechanic, penalization appears to be an ineffective mechanic, whose presence feels detrimental to the experience.

Regarding the two open questions, which covered respectively issues encountered during the experiment and suggestions and comments for improvements, we performed open coding on the answers collected: starting from the pool of 199 answers, we removed 100 answers from the former's pool, as they either mentioned no issue or had no explanation for the issue, and 82 from the latter's pool, as they contained no actually meaningful comment.

We present in Figures 11 and 12 the distribution of topics identified in the answers given to the open-ended questions after our open coding operation and the removal of the answers with no meaningful content.

Regarding issues, the most commonly found topic (40 answers) pertains to bugs or limitations of the experiment: such undesired behaviors are related to the prototypical status of the tool. Additionally, 35 students reported issues related to problems found with the rule-checking feature of the tool and the corresponding feedback mechanic, while 21 students mentioned problems with the external libraries used to implement the BPMN modeling canvas. Lastly, 3 students mentioned problems with the experiment's timing mechanism, which will not be included in future applications of the tool.

For what concerns the question asking for suggestions, instead, the most commonly mentioned topic consisted of suggesting the implementation

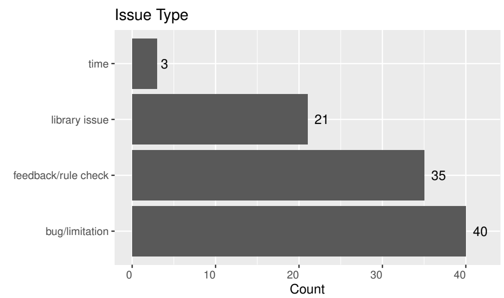


Fig. 11 Topics found in the issues described by the students

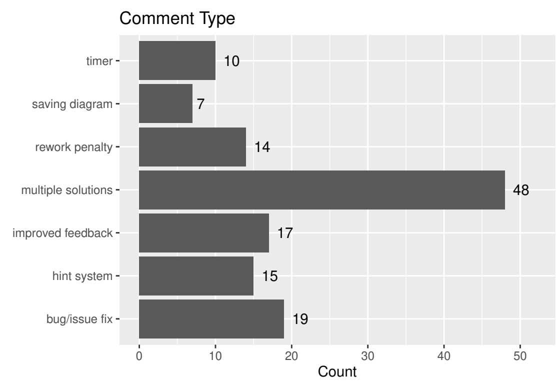


Fig. 12 Topics found in the open comments left by the students

of multiple solutions, with 48 students mentioning such a topic in their answer; other common answers suggested the implementation of a hint system (15 answers), reworking the penalization mechanism by making it less strict (14 answers), and improving the feedback mechanic by having it offer more detailed explanations (17 answers). The remaining answers covered topics that are due to the nature of the experiment (presence of bugs, concerns related to the timer, inability to save and restore modeled diagrams), which are going to be addressed and resolved in time for the actual usage of the tool in a classroom environment.

Answer to RQ4: Students mainly appreciated the *Feedback* and *Progress* mechanics, as shown by the distribution of answers to the questionnaire. The lower appreciation of the *Rewards* mechanic, as well as the dislike of the *Penalty* mechanic is connected to the way the tool evaluates diagrams. Answers to the open questions confirm this issue, as many of them touch on this specific topic.

6 Discussion

For this study, we performed an experiment to assess whether using gamification can be an effective strategy for improving motivation in learning BPMN modeling. To this end, we developed a gamified modeling tool, applied it in a modeling task, and evaluated the diagrams produced by the participants to determine whether there were any improvements in terms of productivity and correctness of the diagram. Moreover, we also evaluated the perception of the gamified tool using the GAMEX questionnaire and the enjoyment and perceived usefulness of the experience with a separate set of questions.

Regarding the size of submissions, we found that no statistically significant change was observed when using the gamified tool compared to the vanilla version. The mechanics inserted in the tool did not motivate students to perform significant changes to their diagrams' sizes by neither adding nor removing elements. The analysis showed, however, that there was an impact on diagram size depending on the exercise under test, meaning that one exercise was easier than the other, as it required fewer elements to be modeled in a correct way.

Regarding the number of submissions, however, gamification actually brought a statistically significant increase. This was an expected result, as the gamified version of the tool tied the mechanics of feedback, rewards, and penalization to a step that checks for correctness on submission. Therefore such activity appeared more appealing. Conversely, the non-gamified version only offered a check on the syntactical correctness of the diagram – without any reward, explanation, or penalty – ; this feature led to a much smaller number of submissions.

Gamification proved to have no statistically significant impact on the syntactical correctness of the diagrams modeled by students: the fact that a syntax check was present in both versions of the application is likely an explanation for this result. There was also no impact on this metric caused by either the order of tasks or the exercises: we can thus say that being able to follow syntax rules when modeling BPMN diagrams is something that depends on individual skills.

Regarding semantic correctness, we found that gamification brought a statistically significant increase when gamification was employed. This is in line with the way the experiment was conducted, as the gamified version of the tool included detailed feedback on the *parts* of the process that had to be modeled.

It is important to mention, however, that the mean values obtained for semantic correctness were quite low for both the vanilla and the gamified version of the tool: this is connected with the way the evaluation engine, responsible for the rules behind the semantic correctness, has been developed. The engine evaluates diagrams depending on whether they contain expected elements, with specific properties, and does not consider elements correct if there's at least one property that is not respected. In practice, non-orthodox solutions are deemed incorrect. The engine does not support different but still correct ways to model the same process.

To address this limitation, we plan in the next version of the tool to modify the engine so that it can identify incorrect constructs and penalize those errors, rather than focus on the similarity to a limited reference set of correct solutions. Such a strategy could also be more feasible, as we expect the common errors for a specific process model to be easier to define rather than all the possible alternative correct models.

Additionally, the analysis showed that the two exercises also had an impact on semantic correctness: this, together with the similar impact on diagram size, confirms indeed that one exercise was easier than the other. We found that the order of tasks had no impact on any of the four metrics we analyzed for RQ1 and RQ2, which means that there was no learning bias, as well as no fatigue effect influencing the results.

Concerning the students' perception of the gamified experience, results were mainly on the neutral side. We consider the absence of negative effects such as stress or frustration when using the tool as favorable results. These results may have been influenced by the fact that the tool was being used in an experimental environment, meaning that continuous usage in a classroom environment may prove to be more effective.

Regarding student enjoyment and perceived usefulness, answers to the questionnaire showed that students appreciated the feedback and progress mechanics present in the tool, while the same cannot be said for the rewarding aspect and the penalization, which were widely disliked. This distribution of answers, as well as the topics found in the answers to the open-ended questions, indicates that the combination of a penalty mechanism with a strict evaluation engine that only accepts a limited set of correct solutions is a severe limitation of the tool, as it makes using the tool frustrating and unfun. Keeping in mind the comments left by students, we can see that the penalty system can be improved with the addition of a hint system or a variation that penalizes only serious mistakes; the implementation of this mechanic is still in its early stages, so there is room for improvement. Another possible solution would be to remove the mechanic entirely.

7 Conclusion and Future Work

In this study, we have investigated whether gamification can be effectively used as a way to improve motivation in learning of BPMN modeling. For this purpose, we developed a web application that included modeling functionalities and gamified mechanics and conducted an experiment with students of a Masters' Degree Information Systems course.

The experiment showed that gamification has a positive impact on the semantic correctness of the solutions modeled by the students, which was expected since the feedback included in the gamified version of the tool aims at improving that specific aspect. Other relevant results show that students are more prone to frequently check the correctness of their diagrams if such an action is tied to in-game rewards, and that visual feedback on diagrams is a well-received addition to a modeling tool.

The tool's appreciation by students was assessed with a questionnaire, that showed that the gamified experience has been perceived in a neutral way (that is, neither exceedingly positive nor negative); we assume that this result is connected to some issues encountered during the experiment setup, as well as to the implementation of some mechanics.

The most pressing issue identified by the students' opinion lies with the way the evaluation engine of the tool is implemented: checking for semantic correctness according only to the presence of specific modeling choices is a limited way to assess diagrams, as multiple solutions can be correct in modeling a specific process. This limitation, coupled with a penalization mechanism for every process part not modeled according to the reference solution, has certainly lowered the enjoyment of the tool, which, for a gamified experience, is a serious issue.

Future plans regarding the tool involve:

- Reworking the evaluation engine so that it allows more correct modeling options and additional variations. Allowing different solutions, as well as different variations of each solution is going to make the experience less frustrating for students, as it will increase their chance of having a correct diagram.

- Improving the penalty mechanism so that it is less strict and is applied only for serious errors and not after every check so that students do not feel demotivated when modeling. Implementing a rewarding mechanism for correcting errors is also something that can be considered, as it would increase the motivation to change and improve a diagram.
- Extending the feedback mechanism so that errors are explained more in detail and are more pertinent to the actual diagram drawn by a student rather than just related to a reference solution. Giving a clear indication of what is wrong and why could facilitate the comprehension of correct modeling practices.
- Developing additional mechanics for long-term usage such as competition in the form of leaderboards, level progression, customization of a student's profile, unlockable rewards, and quest lines.

A second longitudinal experiment with the tool is planned for the new edition of the course, where the tool will be used for the entire duration of the course: using the tool over a longer period will allow a better assessment of whether gamification can improve students' performances.

Declarations

- Competing interests. The authors declare they have no conflicts of interest.

Appendix A Evaluation Criteria

A.1 Example 1

Listing A.1 defines the necessary elements a diagram must have to model correctly the part of the process called *Election Definition*. The criteria involved are **Label**, **Parent**, **Element**, **Target**, and **Pool**.

```
"Task_Election Definition": {
    "Label": ["define", "specify", "date", "time", "call", "appoint"],
    "Parent": {
      "Element": "StartEvent",
      "Label": ["definition","election","manage"],
    }
    "Target": {
      "Element": "ServiceTask",
      "Label": [ "notify", "send", "email", "e-mail", "notification", "mail"]
    },
    "Pool": ["election", "manage"]
}
```

The diagram must contain one *Task* element whose label contains at least one of the following words:

- define;
- specify;
- date;

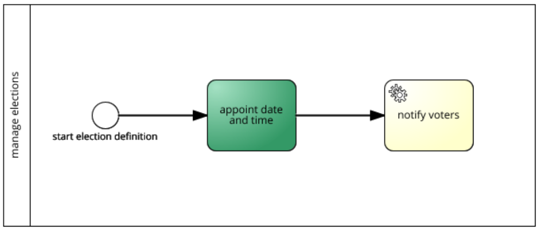


Fig. A1 BPMN diagram containing a Task that satisfies all the specified criteria

- time;
- call;
- appoint.

The Task must also belong to a *Pool* whose name contains at least one between *election* and *manage*.

Parent and **Target** define, respectively, the elements that must come before and after the Task in the diagram: the *Element* attribute of **Parent** specifies that the Task must be preceded by a *Start Event*, and this Event must have a name that contains at least one word among *definition*, *election*, and *manage*. Similarly, the **Target** attribute specified that the Task must have an outgoing sequence flow towards a *Service Task* whose name contains at least one of the strings included in the **Label** array.

Figure A1 shows an example of a diagram containing a Task that satisfies all the criteria specified in Listing A.1.

A.2 Example 2

Listing A.2 shows an example of how criteria **GroupRules**, **EventDefinition**, **MessageSender**, and **MessageTarget** are defined.

```
"GroupRules_Previous Day Reminder": [
{
      "IntermediateCatchEvent": {
        "EventDefinition": "TimerEventDefinition
            ",
        "Parent": {...},
        "Label": [ "day before", "one day before
            ", "a day before", "previous day",
            "day before reservation" ],
        "Pool": [ "reservation" ],
        "Target": {
          "Element": "ServiceTask",
          "Label": [ "reminder", "sms", "unique
              code", "send", "message", "remind
              " ]
        }
    },
      "ServiceTask": {
        "Parent": {
          "Element": "IntermediateCatchEvent",
          "EventDefinition": "
              TimerEventDefinition",
          "Label": [ "day before", "one day
              before", "a day before", "
              previous day", "day before
              reservation" ]
        },
        "Label": [ "reminder", "sms", "unique
            code", "send", "message", "remind"
            ],
        "MessageTarget": {
          "Element": "Participant",
          "Label": [ "sms", "gateway", "message"
               ]
        },
        "MessageSender": {
          "Element": "Participant",
          "Label": [ "sms", "gateway", "message"
               ]
```

```
        },
        "Pool": [ "reservation" ],
        "Target": {...}
      }
  }
]
```

GroupRules specifies multiple, alternate ways to represent the same part of a process: different objects of the array represent groups of BPMN objects that must be present together in the diagram.

In this example, the diagram must contain an *Intermediate Catch Event* that must follow all the conditions on its name, its preceding and following elements in the diagram, and its belonging to a specific pool; in addition to these conditions, the Event must also be an event of type *Timer*.

The presence of an Event that satisfies all of its conditions is not sufficient to consider the corresponding part modeled: the diagram must also contain a *Service Task* that satisfies its own criteria. Among these criteria, the Service Task must also have a message exchange (that is, having both an outgoing and an incoming message flow) with a different *Pool*, identified by the **Participant** attribute. These conditions are specified by the **MessageTarget** and **MessageSender** attributes, which define respectively the element for the outgoing and incoming message flow.

Moreover, it is not enough for a diagram to contain the two elements, since the two of them must also be connected, as shown by the equal constraints of the *Parent* and *Target* attributes in comparison to the original elements.

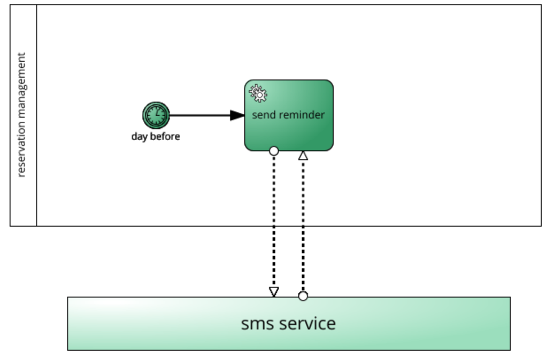


Fig. A2 BPMN diagram containing an Intermediate Timer Event, a Service Task, and a Pool interacting in a correct way

Figure A2 shows an example of a diagram containing an Intermediate Timer Event connected to a Service Task that exchanges messages with an external pool, as specified by Listing A.2.

A.3 Example 3

Listing A.3 shows an example of how criteria **BoundingElement**, **BoundEvent**, and **Interrupting** are defined.

```
"GroupRules_Reservation Cancellation": [
  {
    "BoundaryEvent": {
      "BoundingElement": {
        "Element": "Task",
        "Label": [ "cancel", "delete", "remove
            ", "quit", "stop" ]
      },
      "EventDefinition": "TimerEventDefinition
          ",
      "Interrupting": true,
      "Label": [ "two days", "2 days" ],
      "Pool": [ "reservation" ],
      "Target": {...}
    },
```

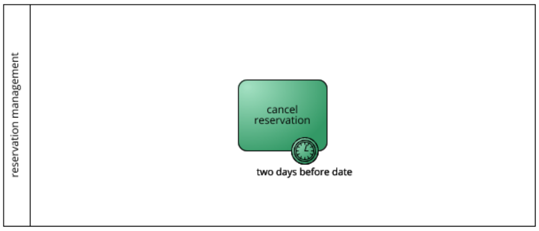


Fig. A3 BPMN diagram containing a Task with a correctly modeled Boundary Interrupting Timer Event

```
    "Task": {
      "BoundEvent": {
        "EventDefinition": "
            TimerEventDefinition",
        "Interrupting": true,
        "Label": [ "two days", "2 days" ]
      },
      "Parent": {...},
      "Label": [ "cancel", "delete", "remove",
           "quit", "stop" ],
      "Pool": [ "reservation" ],
      "Target": {...}
    }
  }
]
```

The example defines the criteria regarding how a specific *Boundary Event* must be modeled: the diagram must contain a *Task* with an attached Boundary Interrupting Timer Event, with **BoundingElement** specifying the conditions for the Event and **BoundEvent** for the Task.

Labels must match for both the elements, and both of them specify that the Event must also be **Interrupting**.

Figure A3 shows a task with a correctly modeled Boundary Interrupting Timer Event, following the criteria defined in Listing A.3.

Appendix B Evaluation Rules

Table B1 Syntax rules which are always checked

Rule Name	Rule Description
PoolNumber	There must be at least two separate, non-empty pools
ManualTaskPresence	Manual tasks must not be present
OnlyTask	Process must not contain only simple tasks from start to end
TaskPresence	At least one simple task must be present in the process
TimerEventPresence	At least one timer event, either intermediate, boundary or start, must be present in the process
EndEventFlow	There cannot be elements that do not have a sequence flow that reaches an end event.
SingleStartEvent	Every pool can have only one start event

Table B2 Syntax rules which are checked depending on the existence of a given element

Rule Name	Rule Description
MessageCatchEvent	Catch message events must have an incoming message flow from another process
MessageThrowEvent	Throw message events must have an outgoing message flow to another process
MessageCatchTask	Catch message tasks must have an incoming message flow from another process
MessageThrowTask	Throw message tasks must have an outgoing message flow to another process
ServiceTaskBoundaryEvent	Service tasks cannot have a boundary event
BlankBoundaryEvent	Boundary events must not be blank
BlankEvent	Intermediate events present in the process must not be blank

Table B3 Syntax rules which are checked depending on the existence of a given element

Rule Name	Rule Description
MessageFlow	Message flows can only be sent from throw message tasks/events, service tasks, or collapsed pools
SignalEventsPresence	If one type of signal event is present, then there must be at least one signal event of the other type
SignalEventsNaming	For every name given to a signal event of a given type, at least one signal event of the other type must exist with the same name
BoundaryEventFlow	The outgoing flow of a boundary event must either reach an end event or reconnect to the original flow
BoundTaskFlow	A task which has a boundary event attached must also have a standard outgoing sequence flow
HumanLane	Lanes must be named after a human actor

Table B4 Rules for the *seaside* exercise

Rule Name	Involved Elements
Reservation Start	Start Event
Registration Start	Start Event
Reservation End	EndEvent
Registration Failure	EndEvent
Registration Data Entry	Task
Verification Result	ExclusiveGateway
Reservation Unconfirmed	EndEvent
Registration Success	EndEvent
Registration Result	ExclusiveGateway
Reservation Confirmation	ExclusiveGateway
Document Verification	Task
Reservation Message	Multiple Elements
Reservation	Task

Table B5 Rules for the *elections* exercise

Rule Name	Involved Elements
Election Start	StartEvent
Election Definition	Task
Election End	EndEvent
Voting Start	StartEvent
Vote Confirmed	EndEvent
Security Code Verification	Multiple Elements
Election Result	Multiple Elements
Email Notification Management	ServiceTask
Security Code Management	Multiple Elements
Signup Result	Multiple Elements
Vote Confirmation	Multiple Element
Voting	Task
Voting Signup	Task

Appendix C Questionnaire

Table C6 GAMEX Questionnaire Q1-2

Question ID	Question Group	Question
Q1.1	Enjoyment	Playing the game was fun.
Q1.2	Enjoyment	I liked playing the game.
Q1.3	Enjoyment	I enjoyed playing the game very much.
Q1.4	Enjoyment	My game experience was pleasurable.
Q1.5	Enjoyment	I think playing the game is very entertaining.
Q1.6	Enjoyment	I would play this game for its own sake, not only when being asked to.
Q2.1	Absorption	Playing the game made me forget where I am.
Q2.2	Absorption	I forgot about my immediate surroundings while I played the game.
Q2.3	Absorption	After playing the game, I felt like coming back to the "real world" after a journey.
Q2.4	Absorption	Playing the game "got me away from it all."
Q2.5	Absorption	While playing the game I was completely oblivious to everything around me.
Q2.6	Absorption	While playing the game I lost track of time.

Table C7 GAMEX Questionnaire Q3-4

Question ID	Question Group	Question
Q3.1	Creative Thinking	Playing the game sparked my imagination.
Q3.2	Creative Thinking	While playing the game I felt creative.
Q3.3	Creative Thinking	While playing the game I felt that I could explore things.
Q3.4	Creative Thinking	While playing the game I felt adventurous.
Q4.1	Activation	While playing the game I felt activated.
Q4.2	Activation	While playing the game I felt nervous.
Q4.3	Activation	While playing the game I felt frenetic.
Q4.4	Activation	While playing the game I felt excited.

Table C8 GAMEX Questionnaire Q5-6

Question ID	Question Group	Question
Q5.1	Possible Negative Effects	While playing the game I felt upset.
Q5.2	Possible Negative Effects	While playing the game I felt hostile.
Q5.3	Possible Negative Effects	While playing the game I felt frustrated.
Q6.1	Dominance	While playing the game I had the feeling of being in charge.
Q6.2	Dominance	While playing the game I felt influential.
Q6.3	Dominance	While playing the game I felt autonomous.
Q6.4	Dominance	While playing the game I felt confident.

Table C9 Gamified mechanics questions

Question ID	Gamified Element	Question
Q7.1	Rewards	Did you appreciate the rewards mechanism (points obtained every time you unlocked a rule)?
Q7.2	Rewards	Did you perceive the rewards mechanism as useful?
Q7.3	Rewards	How influential was the rewards mechanism in the whole experience with the tool?
Q8.1	Penalty	Did you appreciate the penalty mechanism (reduction of the points after each error)?
Q8.2	Penalty	Did you perceive the penalty mechanism as useful?
Q8.3	Penalty	How influential was the penalty mechanism in the whole experience with the tool?
Q9.1	Progress	Did you appreciate the progress mechanism (percentage of progress of the last solution submitted)?
Q9.2	Progress	Did you perceive the progress mechanism as useful?
Q9.3	Progress	How influential was the progress mechanism in the whole experience with the tool?
Q10.1	Feedback	Did you appreciate the feedback mechanism (identification of the found and correct rules with color coding on the diagram)?
Q10.2	Feedback	Did you perceive the feedback mechanism as useful?
Q10.3	Feedback	How influential was the feedback mechanism in the whole experience with the tool?

Open Questions

1. *Did you experience any issues with the tool?*
2. *Do you have any comment or suggestion for tool or game rules improvement?*

Appendix D Exercises

D.1 Exercise 1: Seaside Reservation

Consider the following scenario:

A municipality in a seaside tourist resort wants to develop an information system to manage the reservations of free beaches in its territory in order to comply with the spacing provisions related to the COVID-19 outbreak.

The reservation procedure involves two stages: an initial registration (done once) and then the actual reservation, which can be done multiple times.

In the registration phase, after the citizen enters his/her personal details (including his/her cell phone) and a scan of his/her ID, the system verifies the cell phone. Then a city employee verifies the validity of the document. At this point a confirmation SMS is sent and the reservation can be made.

A reservation involves selecting the days the citizen intends to reserve and the number of people (indicating the number of adults and the number of children under 12).

Reservation is made by a registered user indicating the range of days he/she wants to reserve and the persons, the system checks availability and asks for confirmation, after which it sends a summary SMS. It is possible to cancel the reservation up to two days before the beginning of the booked period. If the reservation has not been canceled, the system, the day before the reservation, sends a reminder SMS with the unique code to be shown to the attendants at the beach entrance. At the end of the booked period, the system sends an SMS asking to fill out a satisfaction questionnaire. The user can fill out the questionnaire by logging into the reservation system.

D.2 Exercise 2: University Elections

Consider the following scenario:

A public university intends to implement a platform to be able to handle various types of online elections among its employees. These may be, for example, elections of department directors, or of a departmental board, or of representatives of the technical category, or of members of the academic senate, and so on.

The Election Commission may call new elections, specifying all the necessary information and the date (and time) when the elections will be held (at least 15 days in advance).

Voters are notified by e-mail and, on the appointed day, can log on to the system to vote. The voting operation consists of choosing, from the eligible members of the department, a number of people equal to the number of preferences allowed. Confirmation is then requested, after which the vote becomes final and can no longer be changed.

The security of voting operations is ensured by two-factor authentication: in addition to providing their password, voters must enter an 8-digit numerical code sent by the system via SMS to the cell phone number registered in the system.

At the end of the voting time, the election commission will display the results: the list of voters, whether a quorum has been reached, and only if a quorum is reached, the list of preferences given to each eligible member. In that list, the employees actually elected are highlighted.

References

[1] Omg, O., Parida, R. & Mahapatra, S. Business process model and notation (bpmn) version 2.0. Object Management Group. **1**, 18 (2011)

[2] Bucchiarone, A., Savary-Leblanc, M., Le Pallec, X., Cicchetti, A., Gérard, S., Bassanelli, S., Gini, F. & Marconi, A. Gamifying model-based engineering: the PapyGame experience. Software And Systems Modeling. (2023,3,4), DOI: https://doi.org/10.1007/s10270-023-01091-8

[3] Jurgelaitis, Mantas, et al. "Implementing gamification in a university-level UML modeling course: A case study." Computer

Applications in Engineering Education 27.2 (2019): 332-343.

[4] Deterding, S., Dixon, D., Khaled, R. & Nacke, L. From Game Design Elements to Gamefulness: Defining "Gamification". Proceedings Of The 15th International Academic MindTrek Conference: Envisioning Future Media Environments. pp. 9-15 (2011), DOI: https://doi.org/10.1145/2181037.2181040

[5] Groh, F. Gamification: State of the art definition and utilization. Institute Of Media Informatics Ulm University. **39** pp. 31 (2012)

[6] Marin, B., Frez, J., Cruz-Lemus, J. & Genero, M. An Empirical Investigation on the Benefits of Gamification in Programming Courses. ACM Trans. Comput. Educ. **19** (2018,11), DOI: https://doi.org/10.1145/3231709

[7] Prabawa, H., Sutarno, H., Kusnendar, J. & Rahmah, F. Learning basic programming using CLIS through gamification. Journal Of Physics: Conference Series. **1013**, 012099 (2018,5), DOI: https://dx.doi.org/10.1088/1742-6596/1013/1/012099

[8] Rojas-López, A. & Rincón-Flores, E. Gamification as Learning Scenario in Programming Course of Higher Education. Learning And Collaboration Technologies. Learning And Teaching. pp. 200-210 (2018)

[9] Ayub, M., Toba, H., Wijanto, M., Yong, S. & Wijaya, B. Gamification for blended learning in higher education. World Transactions On Engineering And Technology Education. **17** pp. 76-81 (2019,1)

[10] Tasadduq, M., Khan, M., Nawab, R., Jamal, M. & Chaudhry, M. Exploring the effects of gamification on students with rote learning background while learning computer programming. Computer Applications In Engineering Education. **29**, 1871-1891 (2021), DOI: https://doi.org/10.1002/cae.22428

[11] Fraser, G. Gamification of Software Testing. 2017 IEEE/ACM 12th International Workshop On Automation Of Software Testing (AST). pp. 2-7 (2017), DOI: https://doi.org/10.1109/AST.2017.20

[12] Bedwell, K., Garaccione, G., Coppola, R., Ardito, L. & Morisio, M. BIPMIN: A Gamified Framework for Process Modeling Education. Information. **14** (2023), https://www.mdpi.com/2078-2489/14/1/3, DOI: https://doi.org/10.3390/info14010003

[13] SAP Signavio SAP Signavio The Only All-in-One Business Process Software. https://signavio.com Accessed on 14/12/2023

[14] Deci, E.; Ryan, R. Intrinsic motivation and self-determination in human behavior. (Springer Science & Business Media,2013)

[15] Sailer, M., Hense, J., Mayr, S. & Mandl, H. How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction. Computers In Human Behavior. **69** pp. 371-380 (2017), DOI: https://doi.org/10.1016/j.chb.2016.12.033

[16] Wee, S. & Choong, W. Gamification: Predicting the effectiveness of variety game design elements to intrinsically motivate users' energy conservation behaviour. Journal Of Environmental Management. **233** pp. 97-106 (2019), DOI: https://doi.org/10.1016/j.jenvman.2018.11.127

[17] Chou, Y. Actionable Gamification: Beyond Points, Badges, and Leaderboards; Createspace Independent Publishing Platform: Scotts Valley, CA, USA, 2015.

[18] Sailer, M. & Homner, L. The Gamification of Learning: a Meta-analysis. Educational Psychology Review. **32**, 77-112 (2020,3,1), DOI: https://doi.org/10.1007/s10648-019-09498-w

[19] Porto, D., Jesus, G., Ferrari, F. & Fabbri, S. Initiatives and challenges of using gamification in software engineering: A Systematic Mapping. Journal Of Systems And Software. **173** pp. 110870 (2021,3,1), DOI: https://doi.org/10.1016/j.jss.2020.110870

[20] Barreto, C. & França, C. Gamification in Software Engineering: A literature Review. 2021 IEEE/ACM 13th International Workshop On Cooperative And Human Aspects Of Software Engineering (CHASE). pp. 105-108 (2021), DOI: https://doi.org/10.1109/CHASE52884.2021.00020

[21] Alhammad, M. & Moreno, A. Gamification in software engineering education: A systematic mapping. Journal Of Systems And Software. **141** pp. 131-150 (2018), DOI: https://doi.org/10.1016/j.jss.2018.03.065

[22] Cursino, R., Ferreira, D., Lencastre, M., Fagundes, R. & Pimentel, J. Gamification in Requirements Engineering: A Systematic Review. 2018 11th International Conference On The Quality Of Information And Communications Technology (QUATIC). pp. 119-125 (2018), DOI: https://doi.org/10.1109/QUATIC.2018.00025

[23] Mäntylä, M. & Smolander, K. Gamification of Software Testing - An MLR. Product-Focused Software Process Improvement. pp. 611-614 (2016), DOI: https://doi.org/10.1007/978-3-319-49094-6_46

[24] Vargas-Enriquez, J., Garcia-Mundo, L., Genero, M. & Piattini, M. A Systematic Mapping Study on Gamified Software Quality. 2015 7th International Conference On Games And Virtual Worlds For Serious Applications (VS-Games). pp. 1-8 (2015), DOI: https://doi.org/10.1109/VS-GAMES.2015.7295763

[25] Dubois, D. & Tamburrelli, G. Understanding Gamification Mechanisms for Software Development. Proceedings Of The 2013 9th Joint Meeting On Foundations Of Software Engineering. pp. 659-662 (2013), DOI: https://doi.org/10.1145/2491411.2494589

[26] Sonarsource S.A. Clean Code - Developer First - Sonar. https://www.sonarsource.com/ Accessed on 09/05/2023

[27] Dicheva, D., Irwin, K. & Dichev, C. Exploring Learners Experience of Gamified Practicing: For Learning or for Fun?. International Journal Of Serious Games. **6**, 5-21 (2019,9), DOI: https://doi.org/10.17083/ijsg.v6i3.299

[28] Uskov, V. & Sekar, B. Gamification of software engineering curriculum. 2014 IEEE Frontiers In Education Conference (FIE) Proceedings. pp. 1-8 (2014), DOI: https://doi.org/10.1109/FIE.2014.7044098

[29] Gasca-Hurtado, G., Gómez-Álvarez, M. & Manrique-Losada, B. Using gamification in software engineering teaching: Study case for software design. New Knowledge In Information Systems And Technologies: Volume 3. pp. 244-255 (2019)

[30] Calderón, A., Ruiz, M. & O'Connor, R. Coverage of the ISO 21500 Standard in the Context of Software Project Management by a Simulation-Based Serious Game. Software Process Improvement and Capability Determination: 17th International Conference, SPICE 2017, Palma de Mallorca, Spain, October 4–5, 2017, Proceedings (pp. 399–412). (2017,9) DOI: https://doi.org/10.1007/978-3-319-67383-7_29

[31] Maxim, B., Kaur, R., Apzynski, C., Edwards, D. & Evans, E. An Agile Software Engineering Process Improvement Game. 2016 IEEE Frontiers In Education Conference (FIE). pp. 1-4 (2016), https://doi.org/10.1109/FIE.2016.7757682

[32] Paiva, J., Haraszczuk, A., Queirós, R., Leal, J., Swacha, J. & Kosta, S. FGPE Gamification Service: A GraphQL Service to Gamify Online Education. Trends And Applications In Information Systems And Technologies. pp. 480-489 (2021), DOI: https://doi.org/10.1007/978-3-030-72654-6_46

[33] Paiva, J., Queirós, R., Leal, J., Swacha, J. & Miernik, F. Managing Gamified Programming Courses with the FGPE Platform. Information. **13** (2022), DOI: https://doi.org/10.3390/info13020045

[34] Kutun, B. & Schmidt, W. BPMN Wheel: Board Game for Business Process Modelling. (2019,10)

[35] Djaouti, D., Alvarez, J. & Jessel, J. Classifying serious games: the G/P/S model. Handbook Of Research On Improving Learning And Motivation Through Educational Games: Multidisciplinary Approaches. pp. 118-136 (2011)

[36] Eclipse Foundation Papyrus https://www.eclipse.org/papyrus/ Accessed on 13/04/2023

[37] Cosentino, V., Gérard, S. & Cabot, J. A Model-based Approach to Gamify the Learning of Modeling. (2017,11)

[38] Marín, B., Larenas, F. & Giachetti, G. Learning Conceptual Modeling Design Through the Classutopia Serious Game. International Journal Of Software Engineering And Knowledge Engineering. **28** pp. 1679-1699 (2018,11)

[39] Júnior, E. & Farias, K. ModelGame: A Quality Model for Gamified Software Modeling Learning. Proceedings Of The 15th Brazilian Symposium On Software Components, Architectures, And Reuse. pp. 100-109 (2021), https://doi.org/10.1145/3483899.3483910

[40] Marín, B. Lessons Learned About Gamification in Software Engineering Education. Research Anthology on Developments in Gamification and Game-Based Learning (2022,1)

[41] Mancebo, J., Garcia, F., Pedreira, O. & Moraga, M. BPMS-Game: Tool for Business Process Gamification. Business Process Management Forum. pp. 127-140 (2017), DOI: https://doi.org/10.1007/978-3-319-65015-9_8

[42] Dumas, M., La Rosa, M., Mendling, J. & A Reijers, H. Fundamentals of business process management. (Springer,2018), DOI: https://doi.org/10.1007/978-3-662-56509-4

[43] SAP Signavio Business Process Model & Notation - BPMN Introductory Guide - Signavio https://www.signavio.com/bpmn-introductory-guide/ Accessed on 08/05/2023

[44] Meta React https://react.dev/ Accessed on 14/04/2023

[45] Camunda Platform bpmn-js: BPMN 2.0 rendering toolkit and web modeler Toolkits bpmn.io https://bpmn.io/toolkit/bpmn-js/ Accessed on 14/04/2023

[46] Camunda Platform bpmn-io/bpmnlint: Validate BPMN diagrams based on configurable lint rules. https://github.com/bpmn-io/bpmnlint Accessed on 14/04/2023

[47] Ferreira Costa, I. & Oliveira, S. The use of gamification to support the teaching-learning of software exploratory testing: an experience report based on the application of a framework. 2020 IEEE Frontiers In Education Conference (FIE). pp. 1-9 (2020), DOI: https://doi.org/10.1109/FIE44824.2020.9273943

[48] Cacciotto, F., Fulcini, T., Coppola, R. & Ardito, L. A Metric Framework for the Gamification of Web and Mobile GUI Testing. 2021 IEEE International Conference On Software Testing, Verification And Validation Workshops (ICSTW). pp. 126-129 (2021), DOI: https://doi.org/10.1109/ICSTW52544.2021.00032

[49] Jedlitschka, A. & Pfahl, D. Reporting guidelines for controlled experiments in software engineering. 2005 International Symposium On Empirical Software Engineering, 2005. pp. 10-pp (2005), DOI: https://doi.org/10.1109/ISESE.2005.1541818

[50] Eppmann, R., Bekk, M. & Klein, K. Gameful Experience in Gamification: Construction and Validation of a Gameful Experience Scale [GAMEX]. Journal Of Interactive Marketing. **43**, 98-115 (2018), DOI: https://doi.org/10.1016/j.intmar.2018.03.002

[51] The R Foundation R: The R Project for Statistical Computing https://www.r-project.org/ Accessed on 15/03/2023

[52] Vegas, S., Apa, C. & Juristo, N. Crossover Designs in Software Engineering Experiments: Benefits and Perils. IEEE Transactions On Software Engineering. **42**, 120-135 (2016), DOI: https://doi.org/10.1109/TSE.2015.2467378

[53] The R Foundation CRAN - Package likert M. Package ‘likert’ https://cran.r-project.org/web/packages/likert/ Accessed on 15/03/2023

[54] Ryan, G. & Bernard, H. Techniques to Identify Themes. Field Methods. **15**, 85-109 (2003), DOI: https://doi.org/10.1177/1525822X02239569

[55] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M., Regnell, B. & Wesslén, A. Experimentation in software engineering. (Springer Science & Business Media,2012)

[56] Jordan, P. Effects of an extrinsic reward on intrinsic motivation: A field experiment. Academy Of Management Journal. **29**, 405-412 (1986), DOI: https://doi.org/10.5465/256195