

Gamification of business process modeling education: an experimental analysis

*Original*

Gamification of business process modeling education: an experimental analysis / Garaccione, Giacomo; Coppola, Riccardo; Ardito, Luca; Torchiano, Marco. - In: SOFTWARE AND SYSTEMS MODELING. - ISSN 1619-1374. - ELETTRONICO. - (2024). [10.1007/s10270-024-01171-3]

*Availability:*

This version is available at: 11583/2989007 since: 2024-05-27T08:48:56Z

*Publisher:*

Springer

*Published*

DOI:10.1007/s10270-024-01171-3

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

Springer postprint/Author's Accepted Manuscript

This version of the article has been accepted for publication, after peer review (when applicable) and is subject to Springer Nature's AM terms of use, but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: <http://dx.doi.org/10.1007/s10270-024-01171-3>

(Article begins on next page)

# Gamification of Business Process Modeling Education: An Experimental Analysis

Giacomo Garaccione<sup>1</sup>, Riccardo Coppola<sup>1</sup>, Luca Ardito<sup>1</sup> and Marco Torchiano<sup>1</sup>

<sup>1</sup>Department of Control and Computer Engineering, Politecnico di Torino, Corso Duca degli Abruzzi, 24, Torino, 10129 , Italy.

Contributing authors: [giacomo.garaccione@polito.it](mailto:giacomo.garaccione@polito.it); [riccardo.coppola@polito.it](mailto:riccardo.coppola@polito.it);  
[luca.ardito@polito.it](mailto:luca.ardito@polito.it); [marco.torchiano@polito.it](mailto: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

052 in a clear and understandable way. Business Pro-  
053 cess Model and Notation (BPMN) is a widely  
054 used notation for understanding and analyzing  
055 organizational processes. This standard uses flow-  
056 charting logic to graphically represent business  
057 processes, detailing the logic behind the proce-  
058 dure at both high and low levels. It is an effective  
059 tool for learning how to model and analyze organi-  
060 zational processes. Computer Engineering curric-  
061 ular in different universities include courses where  
062 BPMN modeling is taught. As other modeling  
063 activities in software engineering, Process Model-  
064 ing is often perceived by students as unappealing  
065 and uninteresting if compared to coding activities  
066 [2]. At Politecnico di Torino, specifically, we have  
067 observed students struggling with the application  
068 of correct BPMN constructs, leading to low grades  
069 in the related tasks. We have thereby investigated  
070 the possible application of instruments to enhance  
071 students' motivation and – as a consequence –  
072 ability to learn. As reported by Jurgelaitis et al.,  
073 Gamification can be considered a positive means  
074 of increasing motivation and students' grades in  
075 software modeling activities [3].

092 Gamification, defined by Deterding et al. [4]  
093 as *the use of game design elements in non-game*  
094 *contexts*, has been used with increasing frequency  
095 in both industrial and academic contexts as a  
096 way to increase user motivation, interest, and par-  
097 ticipation in activities that are generally seen as  
098 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 activ-  
ities, 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' motiva-  
tion and results in modeling activities. The tool  
implements commonly used gamification mechan-  
ics 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 com-  
pares 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 contin-  
uation 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

103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153

154 need for competence, the need for autonomy, and  
 155 the need for social relatedness [14, 15]. Address-  
 156 ing these needs resulted in the creation of specific  
 157 game elements tailored to increase human moti-  
 158 vation; we present below an explanation of the  
 159 elements connected to each need:

- 164 • **Competence.** The experience of gaining mas-  
 165 tery, feeling in control of the environment, and  
 166 being effective in the activity one is perform-  
 167 ing. This need has been addressed by defining  
 168 opportunities for players to earn points, awards,  
 169 and badges; allowing comparisons among play-  
 170 ers through leaderboards is another example of  
 171 how this need has been addressed [15];
- 172 • **Autonomy.** The feeling of being in control of  
 173 one's own goals, rather than feeling compelled  
 174 or controlled in one's behavior. Avatars, diverg-  
 175 ing story paths, and different ways to play have  
 176 been created as a way to address this basic need  
 177 [16].
- 178 • **Social Relatedness.** The need to have a  
 179 close relationship and connection with other  
 180 people. It has been addressed with meaning-  
 181 ful storylines centered around the player and  
 182 cooperative and competitive gameplay [15].

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

186 There are many examples of frameworks for  
 187 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 gamifi-  
 cation in a classroom include lowering the learning  
 curve of complex topics [6] and increasing moti-  
 vation in completing complex tasks [9]. Regarding  
 Software Engineering, there have been many liter-  
 ature 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 Tam-  
 burrelli [25], who describe an experiment using  
 competition mechanics. The experiment focused  
 on software quality and programming best prac-  
 tices 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

205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255

256 experience-based learning and educational gami-  
257 fication to stimulate work and improve participa-  
258 tion. The method works by defining five separate  
259 components named Preparation (learning goals,  
260 competencies that students will be expected to  
261 develop), Design (rules to achieve the game's  
262 goal, game materials and roles, steps for develop-  
263 ing the game), Pilotage (pilot session for testing  
264 the game), Scheduling (improvements based on  
265 the pilot experience), and Assessment (evaluation  
266 of the results), with a step-by-step process that  
267 produces an instrument for each component. A  
268 case study where the PID was used in a session  
269 with software developers and software engineering  
270 students showed positive results in terms of enjoy-  
271 ment and creative thinking, as well as benefits in  
272 terms of learning the theory concepts presented.

284 Calderón et al. [30] present ProDec, a  
285 simulation-based serious game that aims to moti-  
286 vate students to learn the principles of software  
287 project management, following the ISO 21500  
288 guidelines. The game offers a training environment  
289 where learners can familiarise themselves with the  
290 different stages of software development as well  
291 as environments for learning and designing game  
292 scenarios. The game covers five process groups of  
293 the ISO 21500 standard and has 100% coverage on  
294 processes that compose several subject groups of  
295 the aforementioned standard, while some of them  
296 are unfortunately not yet covered, as the authors  
297 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 devel-  
opment: 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 mechan-  
ics include rewards, badges, title changes, and  
experience points. The authors describe the imple-  
mentation of this serious game and mention future  
plans to make it available as a downloadable  
Android application.

Lastly, the Framework for Gamified Program-  
ming Education or FGPE [32] provides the spec-  
ifications for the gamification scheme and the  
exercise definition, a collection of gamified exer-  
cises covering different and popular programming  
languages, software for editing the exercises, and  
an interactive learning environment for the stu-  
dents. 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.

307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357

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

379 An example of a serious game used to teach  
380 students good practices for conceptual modeling is  
381 Classutopia [38], a mobile application that imple-  
382 ments a role-playing game as the context in which  
383 students are encouraged to learn. Students take  
384 part in a quest to protect the world of the game  
385 from an enemy in challenges that consist of cor-  
386 recting defective diagrams, with errors harming  
387 the player and effective corrections hindering the  
388 enemy; visual feedback is also present after eval-  
389 uations, with correct changes and mistakes being  
390 colored green and red, respectively. An empirical  
391 evaluation of the game has been conducted with  
392 students of a Software Engineering course: such  
393 evaluation showed preliminary benefits, with the  
394 authors mentioning, however, that further work is  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408

still needed and that the sample used for the eval-  
uation 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 indi-  
cators, 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 model-  
ing language's conventions), and semantic quality  
(how much the model matches with the domain of  
the challenge). A qualitative evaluation of Mod-  
elGame 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 hav-  
ing 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 com-  
pleting 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.

460 **Semantic Quality** consists of the capacity of  
 461 a BPMN diagram to accurately describe its asso-  
 462 ciated domain. Although there are no established  
 463 guidelines, it is possible to perform a *Validation*  
 464 process, which compares the diagram to how the  
 465 process the diagram is supposed to model is exe-  
 466 cuted in the real world to determine whether the  
 467 diagram is coherent with the real-world scenario  
 468 and whether it includes all the relevant aspects of  
 469 the real process.

476 **Pragmatic Quality** relates to the usability  
 477 of a process. This usability is assessed through  
 478 the *Certification* process, which gauges how eas-  
 479 ily the diagram can be understood, how easily  
 480 it can be modified, and how accurately it rep-  
 481 resents the real-world process it is supposed to  
 482 model. A diagram's pragmatic quality depends on  
 483 details such as its graphical layout, its size, and its  
 484 structural complexity; a diagram's Certification  
 485 depends on how consistent its logical structure is  
 486 with the graphical representation, as well as on  
 487 how labels assigned to elements respect naming  
 488 conventions (e.g. short labels, imperative verbs,  
 489 business objects used as subjects).

### 502 3 Tool Description

503 The tool has been developed as a React-based  
 504 [44] web application that uses the *bpmn.js* [45]  
 505 JavaScript library to allow the creation of BPMN  
 506 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.

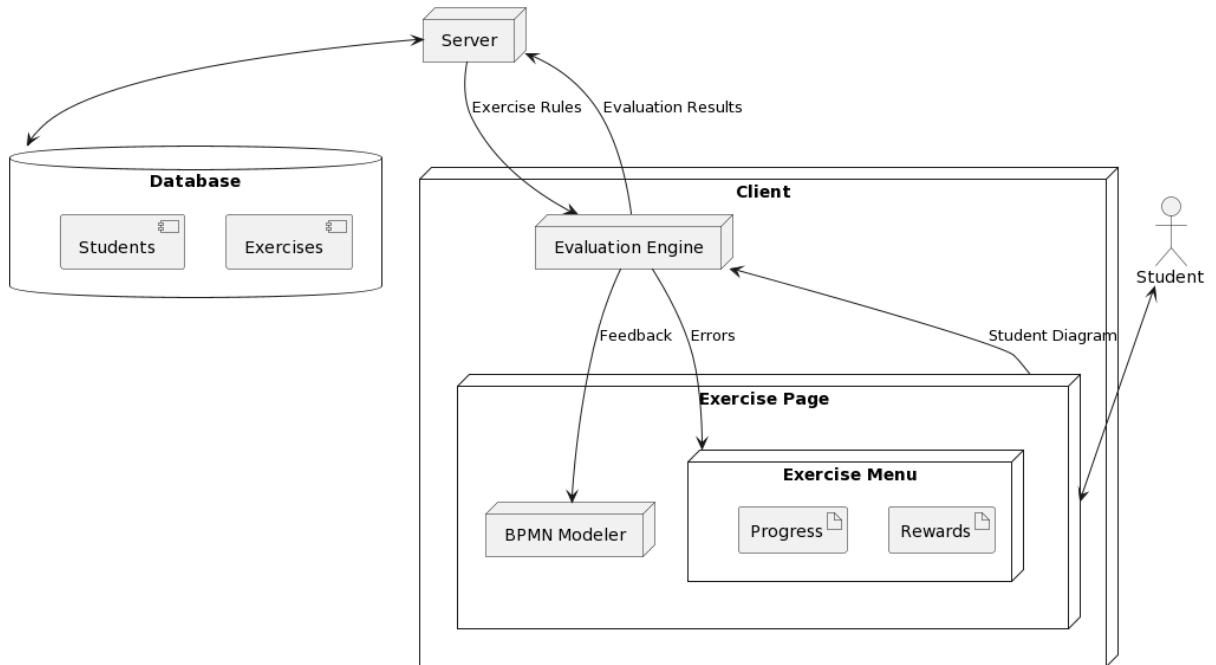
Figure 1 shows the different components that  
 make up the tool: students interact with the client  
 through their web browser and access an exercise  
 page where they can model a diagram that aims to  
 solve the exercise. Every evaluation is saved to the  
 database by sending data to the connected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

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

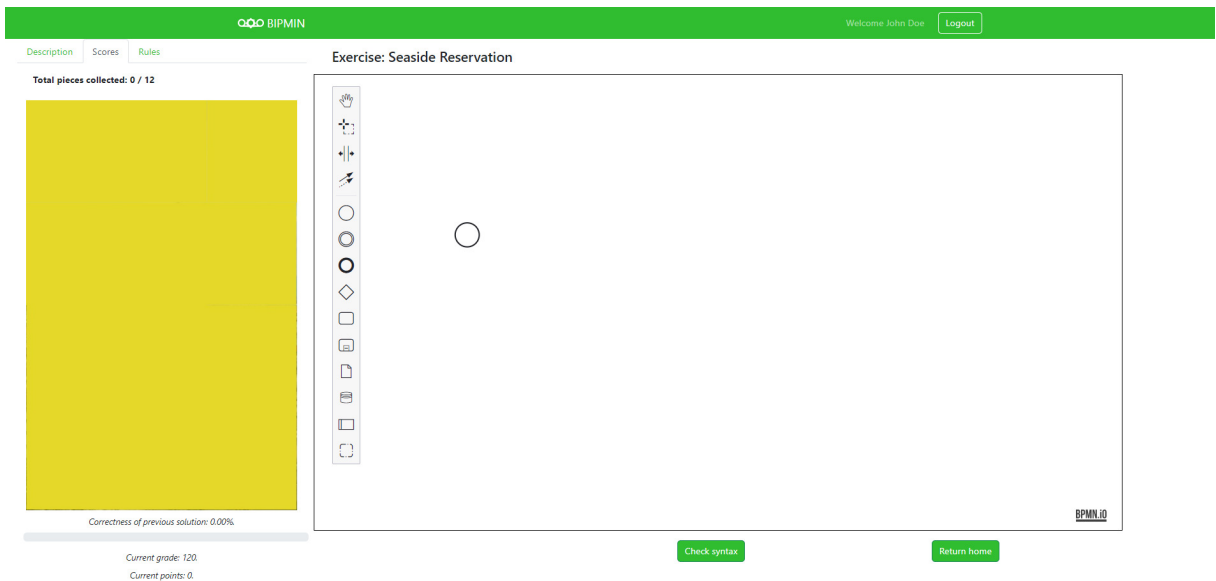
The interface of the tool that allows mod-  
 eling diagrams, which is shown in Figure 2, is  
 divided into a modeling canvas on which students  
 can draw their diagrams using the different ele-  
 ments offered by the BPMN standard, and the  
 tool menu, present on the left-hand side, which  
 contains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

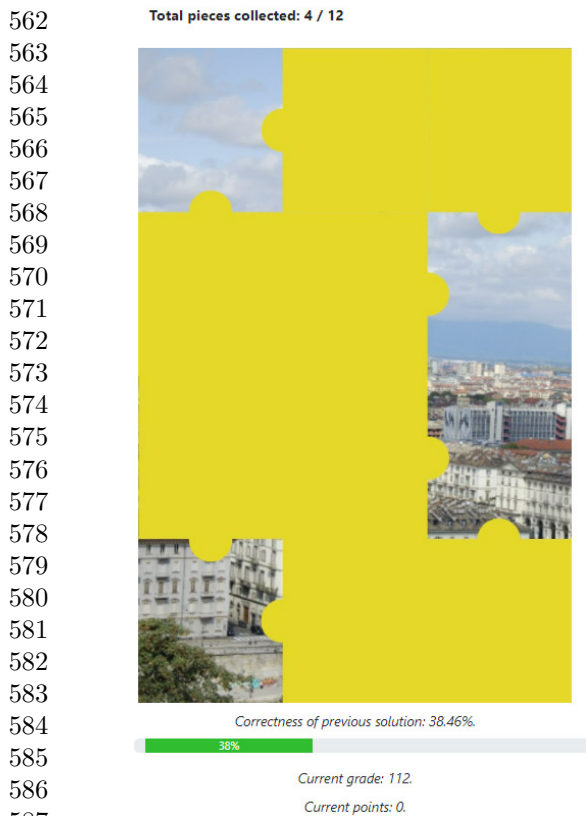
### 3.1.1 Rewards

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

511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561



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

590  
591  
592 pieces of a jigsaw shown in the tool menu, as  
593 shown in Figure 3. The amount of available points  
594 is specified by the *Current points* label and is  
595 updated any time the student either spends points  
596 or obtains new ones.

600 This mechanic is tied to the Core Drive named  
601 *Ownership and Possession* of the Octalysis frame-  
602 work. The concept of rewarding players for well-  
603 performed activities is quite common in gamified  
604 tools: an example of using rewards in a gamified  
605 tool comes from Ferreira and Oliveira [47], who  
606 describe an experiment using a tool for gamify-  
607 ing the teaching of exploratory testing. The tool's  
608  
609  
610  
611  
612

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.

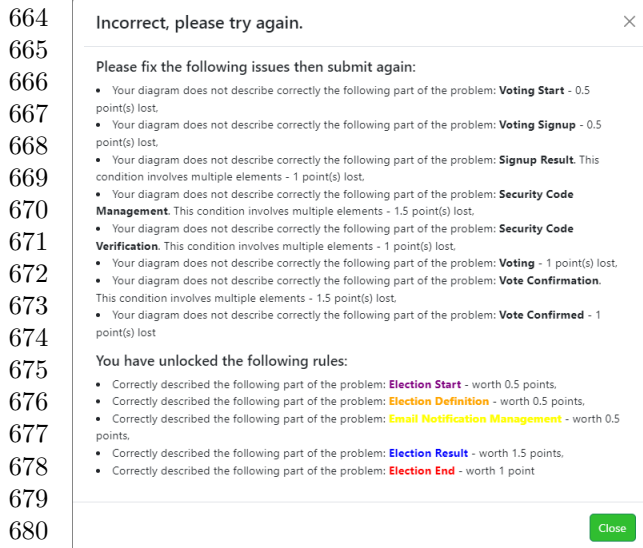
This mechanic is tied to Octalysis' *Accomplishment* Core Drive, as it represents a visual indicator of a student's progression towards successful completion of a modeling assignment: increasing the progress towards success motivates students to try new modeling strategies.

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

613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663



681 **Fig. 4** Feedback modal showing *parts* of the process mod-  
 682 eled correctly and incorrectly

683  
 684  
 685  
 686  
 687 correct; an example of how the modal looks like is  
 688 displayed in Figure 4.

689  
 690 The rules that appear colored in the modal  
 691 window also have their corresponding section of  
 692 the diagram colored in the same way, clearly  
 693 highlighting which element, or combination of  
 694 elements, corresponds to which *part* of the process.

695  
 696 Figure 5(a) shows an example of a diagram  
 697 submitted to a correctness check, while Figure  
 698 5(b) shows the same diagram with colored ele-  
 699 ments, indicating which *parts* of the process the  
 700 student has modeled correctly; these *parts* also  
 701 remain unlocked and have a dedicated section  
 702 in the menu where the student can review them  
 703 again. Each unlocked *part* is also highlighted  
 704 with the same color as in the feedback modal  
 705 and diagram, further reinforcing the mapping and

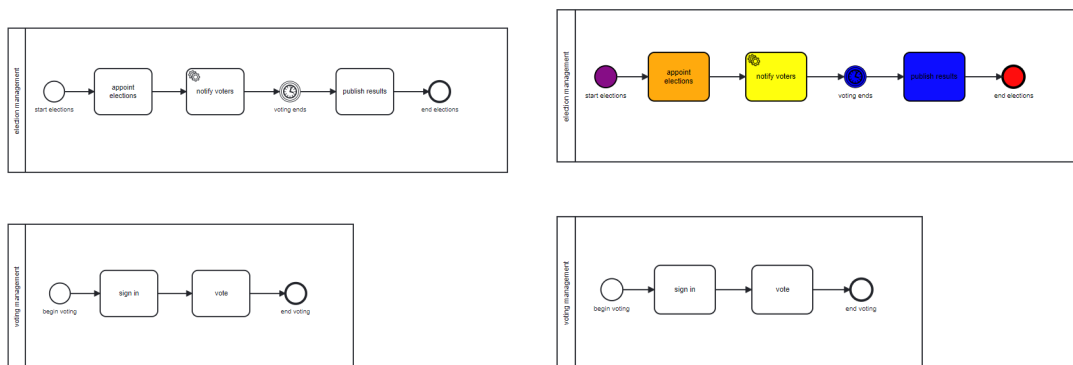
allowing for an easier understanding of the pro-  
 cess. Moreover, the feedback modal also lists the  
 points associated with each *part*, directly connect-  
 ing back to the *Rewards* and *Penalty* mechanics.

This mechanic is connected to the *Empower-  
 ment 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 gami-  
 fied version of the tool makes use of an evaluation  
 engine directly implemented in the tool that eval-  
 uates 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 corre-  
 sponding part of the process modeled correctly.  
 For example, the *part* of the process identified by  
*StartEvent.Reservation Start* is judged as mod-  
 eled 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.

715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765

766 6. **EventDefinition.** This criterion is used only  
 767 for elements that correspond to Events and  
 768 defines a string that specifies which type  
 769 (among, for example, Timer or Message) the  
 770 Event must be for the criterion to be satisfied.  
 771  
 772 7. **MessageTarget.** This criteria defines an  
 773 object that represents another element that  
 774 must exist in the diagram, with its own set  
 775 of criteria. The criterion is verified if there is  
 776 at least one element that satisfies all the inner  
 777 criteria and receives a message flow from the  
 778 original element.  
 779  
 780 8. **MessageSender.** This criteria can be thought  
 781 of as a combination of criteria **MessageTarget**  
 782 and **Parent**, as it defines an object correspond-  
 783 ing to an element with its own set of criteria  
 784 to satisfy that must also have an outgoing mes-  
 785 sage flow towards the element associated with  
 786 the rule to evaluate.  
 787  
 788 9. **BoundingElement.** This criterion is used  
 789 only by elements of type *BoundaryEvent*, and  
 790 defines an object which specifies the element to  
 791 which the event must be bound. There must be  
 792 an element that satisfies all the inner criteria  
 793 and also has a boundary event to satisfy this  
 794 criterion.  
 795  
 796 10. **Interrupting.** This criterion is used only when  
 797 referring to *BoundaryEvents*, and it defines  
 798 a boolean value that specifies whether the  
 799 event is a boundary-interrupting event or a  
 800 non-interrupting one.  
 801  
 802  
 803  
 804  
 805  
 806  
 807  
 808  
 809  
 810  
 811  
 812  
 813  
 814  
 815  
 816

11. **BoundEvent.** This criterion defines an object  
 that specifies the various criteria that a *Bound-*  
*aryEvent* 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 Syn-  
 tactic 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 limita-  
 tions, as unpredictable factors such as human  
 errors in naming elements make defining all possi-  
 ble 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),

817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867

868 **Table 3** Experiment Design

	Task 1		Task 2		
	Exercise	Tool	Exercise	Tool	
871	Group 1	Seaside	Vanilla	Elections	Gamified
872	Group 2	Seaside	Gamified	Elections	Vanilla
873	Group 3	Elections	Vanilla	Seaside	Gamified
874	Group 4	Elections	Gamified	Seaside	Vanilla

875  
876  
877  
878 and allows users of its BPMN modeler to check  
879 whether their diagrams are following basic BPMN  
880 syntax rules. It was for this reason that we decided  
881 to compare a version similar to the one stu-  
882 dents were used to (a BPMN modeler that offers  
883 only syntax evaluations) with a gamified version  
884 that offered more detailed feedback and additional  
885 mechanics.

891 All the participants received both treatments  
892 and had to perform both exercises in two con-  
893 secutive tasks (periods), resulting in two pos-  
894 sible sequences: Gamified-Vanilla and Vanilla-  
895 Gamified. In addition, there are two possible  
896 orders of exercises (objects) for a total of four  
897 different groups. The experimental design is sum-  
898 marized in Table 3. The participants were assigned  
899 randomly to the four groups.

906 After the second exercise session, the partici-  
907 pants were asked to answer a questionnaire about  
908 their experience, including the GAMEX ques-  
909 tionnaire [50] and other questions regarding the  
910 different gamified mechanics. The objective of the  
911 questionnaire was to investigate the perception of  
912 the gamified experience, as well as the influence  
913 the various mechanics had on the experience itself.

### 4.3 Operationalization of Variables

Our experimental design is based on three inde-  
pendent variables. One main factor (*Treatment*)  
corresponds to the testing tool and its two pos-  
sible uses: with or without gamification (Gam-  
ified/Vanilla). Two controlled co-factors related  
respectively to the *Exercises* used (see section  
4.5) and the *Order* in which the tasks are per-  
formed (Vanilla-Gamified or Gamified-Vanilla).  
The dependent variables evaluated in the experi-  
ment 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, includ-  
ing 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, com-  
puted 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 ver-  
sions, 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

919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969

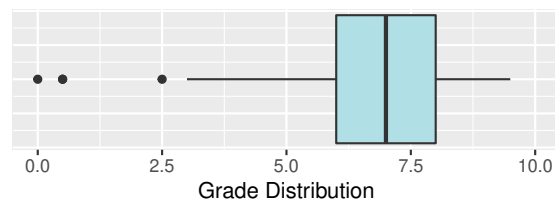
970 questions, asking for the student's opinion regard-  
 971 ing the specific mechanic's influence on the exper-  
 972 ience, its perceived usefulness, and how much  
 973 the student had appreciated the presence of the  
 974 mechanic.

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

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

#### 999 4.4 Participants and Sampling

1000 We recruited participants for the experiment via  
 1001 convenience sampling. All the participants were  
 1002 students enrolled in the master's level in manage-  
 1003 ment engineering *Information Systems* course at  
 1004 Politecnico di Torino. The students were encour-  
 1005 aged to participate in the experimentation by  
 1006 assuring them 2 additional points (out of 30) for  
 1007 the final grade of the exam. Participants were a  
 1008 total of 200 students, with 105 male students and  
 1009 95 female students. We clarified to the students  
 1010 that participation in the experiment was optional  
 1011 since the very beginning of the course and that



**Fig. 6** Distribution of points for BPMN tasks in exams

there would not have been any negative conse-  
 quence for skipping the experiment. Therefore, we  
 can reasonably assert that the students not par-  
 taking in the experiments did not perceive any  
 negative feeling such as being left out or *FOMO*  
 (Fear of Missing Out).

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 model-  
 ing, with the mean value being 7 points out of 10  
 total.

#### 4.5 Experimental subject exercises

The two exercises selected for the experiment con-  
 sist of two descriptions of real-life situations that  
 made use of an Information System; the two exer-  
 cises 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<sup>1</sup>.

## 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<sup>2</sup>; 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<sup>3</sup>. 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.

<sup>1</sup><https://doi.org/10.6084/m9.figshare.22811123.v3>

<sup>2</sup><https://hub.docker.com/r/giacomogaraccione/bipmin>

<sup>3</sup><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 <i>Size</i> of the diagrams submitted by students
$H_{a_{s_0}}$	Gamification has no statistically significant impact on the <i>Number of Submissions</i> made by students
$H_{s_{yc_0}}$	Gamification has no statistically significant impact on the <i>Syntactic Correctness</i> of the diagrams submitted by students
$H_{s_{ec_0}}$	Gamification has no statistically significant impact on the <i>Semantic Correctness</i> 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.

## 1072 4.8 Analysis Method

1073  
1074 We adopt a non-parametric approach to test the  
1075 hypotheses since we expect most variables to be  
1076 not normally distributed. All statistical analysis is  
1077 performed using the statistics tool, R [51].

1080 Following the recommendations given by Vegas  
1081 et al. [52] for analyzing the results of experiments  
1082 whose design is a full factorial crossover, we ana-  
1083 lyzed the data using a repeated measures linear  
1084 mixed model also considering the exercise and  
1085 order (vanilla-gamified or gamified-vanilla) design  
1086 factors to deal with the possible threats to validity  
1087 deriving from the design.

1094 Regarding RQ3 and RQ4, which were evalu-  
1095 ated using a questionnaire with answers that are  
1096 not easily analyzable using statistical hypothesis  
1097 testing, we opted instead for descriptive statistics,  
1098 reporting the results using stacked diverging bar  
1099 charts [53] for the Likert-scale questions. More-  
1100 over, since RQ4 involves open-ended questions, we  
1101 performed open coding [54] on the answers given  
1102 by students, analyzing the content of each answer  
1103 and identifying themes based on repeated words,  
1104 common meanings, and contextual information.

## 1116 4.9 Threats to Validity

1117  
1118 We discuss here the potential threats to the  
1119 study's validity according to the four categories  
1120 defined by Wohlin et al. [55].

**Threats to Internal Validity** concern inter-  
nal 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 consec-  
utively, each with its own time limit. We can thus  
expect the presence of fatigue effects on the partic-  
ipants. 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 differ-  
ent 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 per-  
formed 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 familiar-  
ity 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

1174 amount of time. However, time spent does not nec-  
 1175 essarily correspond to a correct diagram, so we did  
 1176 not include this measure in the analysis.

1177 Regarding *Correctness* metrics, there is no  
 1178 guarantee that the rules we selected were the most  
 1179 effective for measuring a diagram’s correctness.  
 1180 For example, semantic correctness was measured  
 1181 by looking for elements satisfying properties such  
 1182 as the name, specific words, or being connected  
 1183 to other elements. There could have been different  
 1184 ways to measure this correctness, such as look-  
 1185 ing for elements modeled in a way deemed wrong.  
 1186 Moreover, the set of accepted modeling options is  
 1187 not exhaustive, as it does not cover every possible  
 1188 correct alternative way to model a specific part of  
 1189 one of the processes under test.

1200 **Threats to Conclusion Validity** concern  
 1201 the ability to conclude from the results of the  
 1202 study. We employed nonparametric statistical  
 1203 tests that have essentially no statistical prereq-  
 1204 uisite. All measures were collected automatically,  
 1205 meaning that we expect no human errors to  
 1206 impact their collection.

1212 Since the experiment was designed around two  
 1213 different operational environments, the partici-  
 1214 pants were randomly assigned to the four groups,  
 1215 with no statistical difference among the groups  
 1216 regarding BPMN modeling expertise. The random  
 1217 allocation is a confounding factor that could have  
 1218 influenced the statistical results, as a different allo-  
 1219 cation could have led to different statistical test

**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

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.

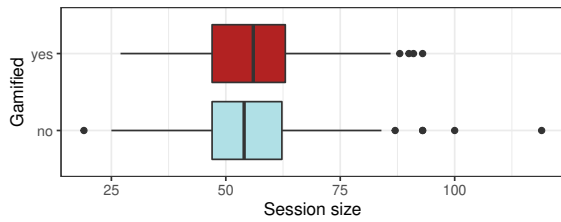
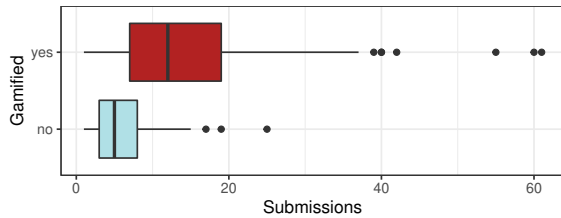
## 5 Results

### 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 ( $H_{s_0}$ )		Submissions ( $H_{as_0}$ )	
	coeff.	p-value	coeff.	p-value
Treatment	-0.3775	0.2170	-4.215	< <b>2e-16</b>
Exercise	-5.9700	< <b>2e-16</b>	-0.2100	0.2525
Order	0.4750	0.4230	-2.2300	<b>0.0056</b>

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

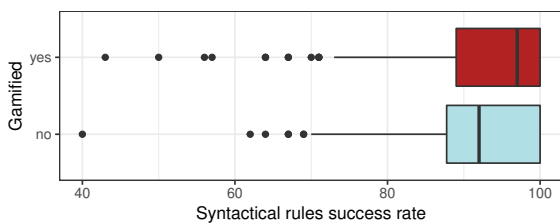
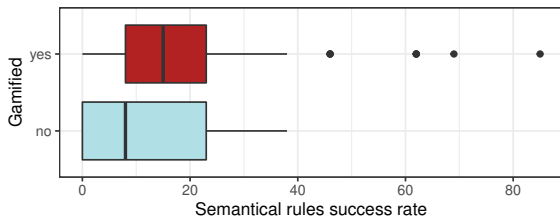
In Table 7 we report the mean, median, and standard deviation for the metrics selected to answer RQ2, e.g. the syntactic and the semantic correctness of each final diagram submitted by students during each exercise. Figure 8 shows box plots for the distribution of each metric aggregated by treatment.

**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 ( $H_{syco}$ ) coeff.	Syntactical ( $H_{syco}$ ) p-value	Semantic ( $H_{sec0}$ ) coeff.	Semantic ( $H_{sec0}$ ) p-value
Treatment	-0.5250	0.2480	-2.0170	<b>0.0211</b>
Exercise	0.3650	1.0000	-2.0980	<b>&lt; 2e-16</b>
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 semantic 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.

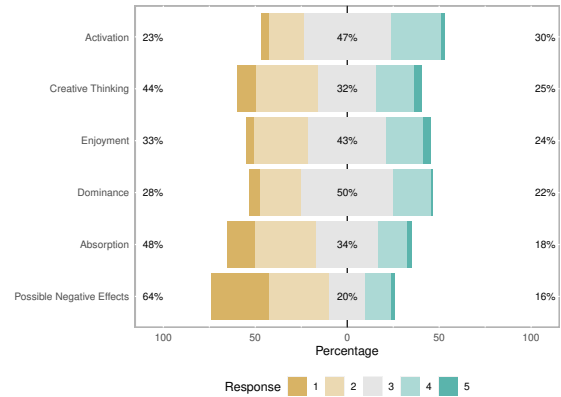
We observe that the order had no significant effect on either syntactical or semantic correctness: we can deduce that changing from a non-gamified version of the tool to a gamified one or vice-versa does not impact students' performances in a significant way, in comparison to using the gamified version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



**Fig. 9** Distribution of the answers to the GAMEX questionnaire

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).

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

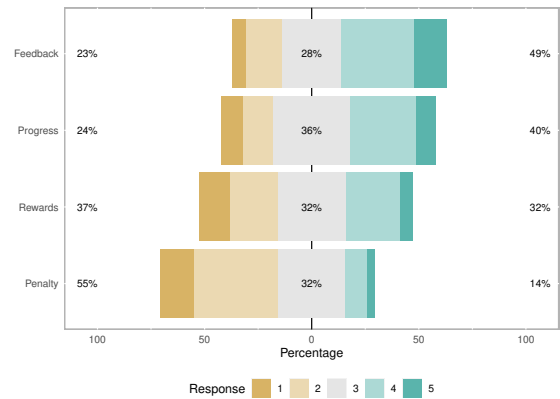
1378 immersed in the gamified experience: it is rea-  
 1379 sonable to assume that a laboratory experience  
 1380 may not be exactly the most immersive environ-  
 1381 ment, and thus participants may not have felt  
 1382 particularly immersed in the tool.  
 1383 particularly immersed in the tool.

1386 A positive result, however, is the one achieved  
 1387 by the *Possible Negative Effects* group: only a  
 1388 total of 16% (14% *Agree* + 2% *Strongly Agree*) of  
 1389 students have answered with an average score that  
 1390 indicates a negative or frustrating experience, pos-  
 1391 sibly implying that the gamified experience, while  
 1392 not exactly immersive, can be thought of as not  
 1393 frustrating, at the very least, which is a necessity  
 1394 for a gamified tool that aims to make unappealing  
 1395 activities more enjoyable.

1404 **Answer to RQ3:** The answers to the GAMEX  
 1405 questionnaire show that students did not perceive  
 1406 the gamified experience as a particularly engaging  
 1407 one. The most encouraging result is a lack of neg-  
 1408 ative effects, while other GAMEX aspects show a  
 1409 distribution of answers more on the neutral side.

## 1413 5.4 RQ4 - Enjoyment and Perceived 1414 Usefulness

1418 The distribution of answers to the questions  
 1419 related to gamified mechanics can be seen in  
 1420 Figure 10, which reports the distribution of  
 1421 answers to the questionnaire; as a way to rep-  
 1422 resent the results in a more compact format, as  
 1423 well as to facilitate discussion, we considered, for  
 1424 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

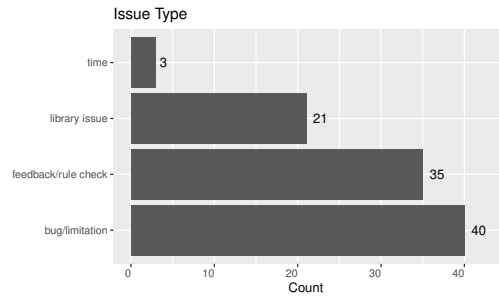
14% strongly disagreeing) of the students did not appreciate the mechanic, resulting in an even split of answers between positive, negative and neutral.

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.

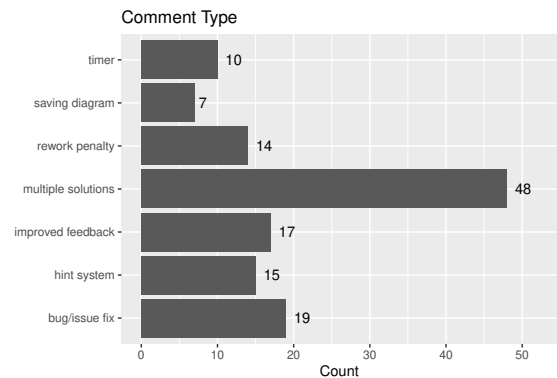
Regarding the two open questions, which covered respectively issues encountered during the experiment and suggestions and comments for improvements, we performed open coding on the answers collected: starting from the pool of 199 answers, we removed 100 answers from the former's pool, as they either 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



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



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

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

1480 of multiple solutions, with 48 students mention-  
 1481 ing such a topic in their answer; other common  
 1482 answers suggested the implementation of a hint  
 1483 system (15 answers), reworking the penalization  
 1484 mechanism by making it less strict (14 answers),  
 1485 and improving the feedback mechanic by having  
 1486 it offer more detailed explanations (17 answers).  
 1487 The remaining answers covered topics that are due  
 1488 to the nature of the experiment (presence of bugs,  
 1489 concerns related to the timer, inability to save and  
 1490 restore modeled diagrams), which are going to be  
 1491 addressed and resolved in time for the actual usage  
 1492 of the tool in a classroom environment.

1502 **Answer to RQ4:** Students mainly appreciated  
 1503 the *Feedback* and *Progress* mechanics, as shown by  
 1504 the distribution of answers to the questionnaire.  
 1505 The lower appreciation of the *Rewards* mechanic,  
 1506 as well as the dislike of the *Penalty* mechanic is  
 1507 connected to the way the tool evaluates diagrams.  
 1508 Answers to the open questions confirm this issue,  
 1509 as many of them touch on this specific topic.

## 1514 6 Discussion

1515 For this study, we performed an experiment to  
 1516 assess whether using gamification can be an effec-  
 1517 tive strategy for improving motivation in learning  
 1518 BPMN modeling. To this end, we developed a  
 1519 gamified modeling tool, applied it in a model-  
 1520 ing task, and evaluated the diagrams produced  
 1521 by the participants to determine whether there  
 1522 were any improvements in terms of productiv-  
 1523 ity and correctness of the diagram. Moreover, we

also evaluated the perception of the gamified tool  
 using the GAMEX questionnaire and the enjoy-  
 ment 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 nei-  
 ther adding nor removing elements. The analysis  
 showed, however, that there was an impact on dia-  
 gram 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, how-  
 ever, gamification actually brought a statistically  
 significant increase. This was an expected result,  
 as the gamified version of the tool tied the mechan-  
 ics 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



1582 found in the answers to the open-ended questions,  
 1583 indicates that the combination of a penalty mech-  
 1584 anism with a strict evaluation engine that only  
 1585 accepts a limited set of correct solutions is a severe  
 1586 limitation of the tool, as it makes using the tool  
 1587 frustrating and unfun. Keeping in mind the com-  
 1588 ments left by students, we can see that the penalty  
 1589 system can be improved with the addition of a  
 1590 hint system or a variation that penalizes only seri-  
 1591 ous mistakes; the implementation of this mechanic  
 1592 is still in its early stages, so there is room for  
 1593 improvement. Another possible solution would be  
 1594 to remove the mechanic entirely.

1603  
 1604

## 1605 7 Conclusion and Future

1606  
 1607

### 1608 Work

1609

1610 In this study, we have investigated whether gami-  
 1611 fication can be effectively used as a way to improve  
 1612 motivation in learning of BPMN modeling. For  
 1613 this purpose, we developed a web application that  
 1614 included modeling functionalities and gamified  
 1615 mechanics and conducted an experiment with stu-  
 1616 dents of a Masters' Degree Information Systems  
 1617 course.

1623  
 1624

1625 The experiment showed that gamification has  
 1626 a positive impact on the semantic correctness of  
 1627 the solutions modeled by the students, which was  
 1628 expected since the feedback included in the gam-  
 1629 ified version of the tool aims at improving that  
 1630 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 feed-  
 back 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 posi-  
 tive 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 stu-  
 dents' opinion lies with the way the evaluation  
 engine of the tool is implemented: checking for  
 semantic correctness according only to the pres-  
 ence of specific modeling choices is a limited way  
 to assess diagrams, as multiple solutions can be  
 correct in modeling a specific process. This limi-  
 tation, coupled with a penalization mechanism for  
 every process part not modeled according to the  
 reference solution, has certainly lowered the enjoy-  
 ment 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**.

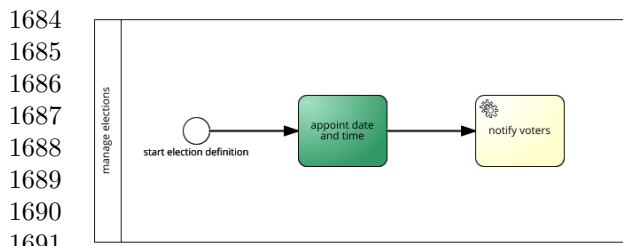
```

1 "Task_Election Definition": {
2   "Label": ["define", "specify", "date", "time",
3     "call", "appoint"],
4   "Parent": {
5     "Element": "StartEvent",
6     "Label": ["definition", "election", "manage"]
7   },
8   "Target": {
9     "Element": "ServiceTask",
10    "Label": ["notify", "send", "email", "email",
11      "notification", "mail"]
12 }

```

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

- define;
- specify;
- date;



1692 **Fig. A1** BPMN diagram containing a Task that satisfies  
1693 all the specified criteria

- 1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701
- 1702 • time;
  - 1703 • call;
  - 1704 • appoint.

1705  
1706  
1707 The Task must also belong to a *Pool* whose  
1708 name contains at least one between *election* and  
1709 *manage*.

1710  
1711 **Parent** and **Target** define, respectively, the  
1712 elements that must come before and after the Task  
1713 in the diagram: the *Element* attribute of **Par-**  
1714 **ent** specifies that the Task must be preceded by a  
1715 *Start Event*, and this Event must have a name that  
1716 contains at least one word among *definition*, *elec-*  
1717 *tion*, and *manage*. Similarly, the **Target** attribute  
1718 specified that the Task must have an outgoing  
1719 sequence flow towards a *Service Task* whose name  
1720 contains at least one of the strings included in the  
1721 **Label** array.

1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734

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

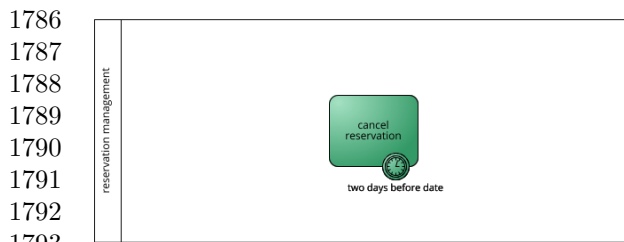
## A.2 Example 2

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

```

1 "GroupRules_Previous Day Reminder": [
2 {
3     "IntermediateCatchEvent": {
4         "EventDefinition": "TimerEventDefinition",
5         "Parent": {...},
6         "Label": [ "day before", "one day before",
7                 "a day before", "previous day",
8                 "day before reservation" ],
9         "Pool": [ "reservation" ],
10        "Target": {
11            "Element": "ServiceTask",
12            "Label": [ "reminder", "sms", "unique
13                    code", "send", "message", "remind
14                    " ]
15        }
16    },
17    "ServiceTask": {
18        "Parent": {
19            "Element": "IntermediateCatchEvent",
20            "EventDefinition": "TimerEventDefinition",
21            "Label": [ "day before", "one day
22                    before", "a day before", "
23                    previous day", "day before
24                    reservation" ]
25        },
26        "Label": [ "reminder", "sms", "unique
27                    code", "send", "message", "remind"
28                    ],
29        "MessageTarget": {
30            "Element": "Participant",
31            "Label": [ "sms", "gateway", "message"
32                    ]
33        },
34        "MessageSender": {
35            "Element": "Participant",
36            "Label": [ "sms", "gateway", "message"
37                    ]
38        }
39    }
40 }
  
```





1794 **Fig. A3** BPMN diagram containing a Task with a cor-  
1795 rectly modeled Boundary Interrupting Timer Event  
1796

```

1797
1798 "Task": {
1799   "BoundEvent": {
1800     "EventDefinition": "
1801       TimerEventDefinition",
1802     "Interrupting": true,
1803     "Label": [ "two days", "2 days" ]
1804   },
1805   "Parent": {...},
1806   "Label": [ "cancel", "delete", "remove",
1807     "quit", "stop" ],
1808   "Pool": [ "reservation" ],
1809   "Target": {...}
1810 }
1811 }
1812 ]

```

1815  
1816 The example defines the criteria regarding  
1817 how a specific *Boundary Event* must be mod-  
1818 eled: the diagram must contain a *Task* with  
1819 an attached Boundary Interrupting Timer Event,  
1820 with **BoundingElement** specifying the condi-  
1821 tions for the Event and **BoundEvent** for the  
1822 Task.

1827 Labels must match for both the elements, and  
1828 both of them specify that the Event must also be  
1829 **Interrupting**.

1832 Figure A3 shows a task with a correctly mod-  
1833 eled Boundary Interrupting Timer Event, follow-  
1834 ing the criteria defined in Listing A.3.  
1835  
1836

## 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

183  
183  
183  
184  
184  
184  
184  
184  
184  
184  
184  
185  
185  
185  
185  
185  
185  
185  
185  
185  
185  
186  
186  
186  
186  
186  
187  
187  
187  
187  
187  
187  
187  
187  
187  
188  
188  
188  
188  
188  
188  
188  
188  
188

**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

**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.

## 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 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.



1990 up to two days before the beginning of the booked  
 1991 period. If the reservation has not been canceled,  
 1992 the system, the day before the reservation, sends a  
 1993 reminder SMS with the unique code to be shown  
 1994 to the attendants at the beach entrance. At the  
 1995 end of the booked period, the system sends an  
 1996 SMS asking to fill out a satisfaction questionnaire.  
 1997 The user can fill out the questionnaire by logging  
 1998 into the reservation system.

## 2007 **D.2 Exercise 2: University**

### 2008 **Elections**

2009 Consider the following scenario:

2010 A public university intends to implement a  
 2011 platform to be able to handle various types of  
 2012 online elections among its employees. These may  
 2013 be, for example, elections of department directors,  
 2014 or of a departmental board, or of representatives  
 2015 of the technical category, or of members of the  
 2016 academic senate, and so on.

2017 The Election Commission may call new elec-  
 2018 tions, specifying all the necessary information and  
 2019 the date (and time) when the elections will be held  
 2020 (at least 15 days in advance).

2021 Voters are notified by e-mail and, on the  
 2022 appointed day, can log on to the system to vote.  
 2023 The voting operation consists of choosing, from  
 2024 the eligible members of the department, a num-  
 2025 ber of people equal to the number of preferences  
 2026 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 provid-  
 ing 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 com-  
 mission 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., Basanelli, 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*



- 2092 [13] SAP Signavio SAP Signavio The Only All-  
2093 in-One Business Process Software. [https://](https://signavio.com)  
2094 [signavio.com](https://signavio.com) Accessed on 14/12/2023
- 2095  
2096  
2097 [14] Deci, E.; Ryan, R. Intrinsic motivation  
2098 and self-determination in human behavior.  
2099 (Springer Science & Business Media,2013)  
2100
- 2101 [15] Sailer, M., Hense, J., Mayr, S. & Mandl,  
2102 H. How gamification motivates: An experi-  
2103 mental study of the effects of specific game  
2104 design elements on psychological need satis-  
2105 faction. *Computers In Human Behavior*. **69**  
2106 pp. 371-380 (2017), DOI: [https://doi.org/](https://doi.org/10.1016/j.chb.2016.12.033)  
2107 [10.1016/j.chb.2016.12.033](https://doi.org/10.1016/j.chb.2016.12.033)  
2108  
2109 [16] Wee, S. & Choong, W. Gamification: Pre-  
2110 dicting the effectiveness of variety game  
2111 design elements to intrinsically motivate  
2112 users' energy conservation behaviour. *Journal*  
2113 *Of Environmental Management*. **233**  
2114 pp. 97-106 (2019), DOI: [https://doi.org/10.](https://doi.org/10.1016/j.jenvman.2018.11.127)  
2115 [1016/j.jenvman.2018.11.127](https://doi.org/10.1016/j.jenvman.2018.11.127)  
2116  
2117 [17] Chou, Y. Actionable Gamification: Beyond  
2118 Points, Badges, and Leaderboards; Cre-  
2119 atespace Independent Publishing Platform:  
2120 Scotts Valley, CA, USA, 2015.  
2121  
2122 [18] Sailer, M. & Homner, L. The Gamifica-  
2123 tion of Learning: a Meta-analysis. *Edu-*  
2124 *cational Psychology Review*. **32**, 77-112  
2125 (2020,3,1), DOI: [https://doi.org/10.1007/](https://doi.org/10.1007/s10648-019-09498-w)  
2126 [s10648-019-09498-w](https://doi.org/10.10648-019-09498-w)
- 2127 [19] Porto, D., Jesus, G., Ferrari, F. & Fab-  
2128 bri, S. Initiatives and challenges of using  
2129 gamification in software engineering: A Sys-  
2130 tematic Mapping. *Journal Of Systems And*  
2131 *Software*. **173** pp. 110870 (2021,3,1), DOI:  
2132 <https://doi.org/10.1016/j.jss.2020.110870>
- 2133 [20] Barreto, C. & França, C. Gamification in  
2134 Software Engineering: A literature Review.  
2135 2021 IEEE/ACM 13th International Work-  
2136 shop On Cooperative And Human Aspects  
2137 Of Software Engineering (CHASE). pp. 105-  
2138 108 (2021), DOI: [https://doi.org/10.1109/](https://doi.org/10.1109/CHASE52884.2021.00020)  
2139 [CHASE52884.2021.00020](https://doi.org/10.1109/CHASE52884.2021.00020)
- 2140 [21] Alhammad, M. & Moreno, A. Gamifica-  
2141 tion in software engineering education: A  
2142 systematic mapping. *Journal Of Systems*  
2143 *And Software*. **141** pp. 131-150 (2018), DOI:  
2144 <https://doi.org/10.1016/j.jss.2018.03.065>
- 2145 [22] Cursino, R., Ferreira, D., Lencastre, M.,  
2146 Fagundes, R. & Pimentel, J. Gamification  
2147 in Requirements Engineering: A Systematic  
2148 Review. 2018 11th International Conference  
2149 On The Quality Of Information And Com-  
2150 munications Technology (QUATIC). pp.  
2151 119-125 (2018), DOI: [https://doi.org/10.](https://doi.org/10.1109/QUATIC.2018.00025)  
2152 [1109/QUATIC.2018.00025](https://doi.org/10.1109/QUATIC.2018.00025)



- 2194 [32] Paiva, J., Haraszczuk, A., Queirós, R., Leal,  
2195 J., Swacha, J. & Kosta, S. FGPE Gamifica-  
2196 tion Service: A GraphQL Service to Gamify  
2197 Online Education. Trends And Applications  
2198 In Information Systems And Technologies.  
2199 pp. 480-489 (2021), DOI: [https://doi.org/  
2200 10.1007/978-3-030-72654-6\\_46](https://doi.org/10.1007/978-3-030-72654-6_46)  
2201  
2202  
2203  
2204  
2205
- 2206 [33] Paiva, J., Queirós, R., Leal, J., Swacha, J.  
2207 & Miernik, F. Managing Gamified Program-  
2208 ming Courses with the FGPE Platform.  
2209 Information. **13** (2022), DOI: [https://doi.  
2210 org/10.3390/info13020045](https://doi.org/10.3390/info13020045)  
2211  
2212  
2213  
2214
- 2215 [34] Kutun, B. & Schmidt, W. BPMN Wheel:  
2216 Board Game for Business Process Mod-  
2217 elling. (2019,10)  
2218  
2219  
2220
- 2221 [35] Djaouti, D., Alvarez, J. & Jessel, J. Classify-  
2222 ing serious games: the G/P/S model. Hand-  
2223 book Of Research On Improving Learn-  
2224 ing And Motivation Through Educational  
2225 Games: Multidisciplinary Approaches. pp.  
2226 118-136 (2011)  
2227  
2228  
2229  
2230
- 2231 [36] Eclipse Foundation Papyrus [https:  
2232 //www.eclipse.org/papyrus/](https://www.eclipse.org/papyrus/) Accessed on  
2233 13/04/2023  
2234  
2235  
2236  
2237
- 2238 [37] Cosentino, V., Gérard, S. & Cabot, J.  
2239 A Model-based Approach to Gamify the  
2240 Learning of Modeling. (2017,11)  
2241  
2242  
2243  
2244
- [38] Marín, B., Larenas, F. & Giachetti,  
G. Learning Conceptual Modeling Design  
Through the Classutopia Serious Game.  
International Journal Of Software Engineer-  
ing And Knowledge Engineering. **28** pp.  
1679-1699 (2018,11)
- [39] Júnior, E. & Farias, K. ModelGame:  
A Quality Model for Gamified Soft-  
ware Modeling Learning. Proceedings  
Of The 15th Brazilian Symposium  
On Software Components, Architec-  
tures, And Reuse. pp. 100-109 (2021),  
<https://doi.org/10.1145/3483899.3483910>
- [40] Marín, B. Lessons Learned About Gam-  
ification in Software Engineering Educa-  
tion. 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](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](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/bpmlint: Validate BPMN diagrams based on configurable lint rules. <https://github.com/bpmn-io/bpmlint> 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

- 2296 [54] Ryan, G. & Bernard, H. Techniques to  
2297 Identify Themes. *Field Methods*. **15**, 85-  
2298 109 (2003), DOI: [https://doi.org/10.1177/  
2299 1525822X02239569](https://doi.org/10.1177/1525822X02239569)  
2300  
2301  
2302
- 2303 [55] Wohlin, C., Runeson, P., Höst, M., Ohlsson,  
2304 M., Regnell, B. & Wesslén, A. *Experimentation in software engineering*. (Springer  
2305 Science & Business Media,2012)  
2306  
2307  
2308  
2309
- 2310 [56] Jordan, P. Effects of an extrinsic reward  
2311 on intrinsic motivation: A field experiment.  
2312 *Academy Of Management Journal*. **29**, 405-  
2313 412 (1986), DOI: [https://doi.org/10.5465/  
2314 256195](https://doi.org/10.5465/256195)  
2315  
2316  
2317  
2318  
2319  
2320  
2321  
2322  
2323  
2324  
2325  
2326  
2327  
2328  
2329  
2330  
2331  
2332  
2333  
2334  
2335  
2336  
2337  
2338  
2339  
2340  
2341  
2342  
2343  
2344  
2345  
2346