Gamifying Business Process Modeling Education: A Longitudinal Study

Gamifying Business Process Modeling Education: A Longitudinal Study

Giacomo Garaccione
giacomo.garaccione@polito.it
Politecnico di Torino
Torino, Italy

Riccardo Coppola
riccardo.coppola@polito.it
Politecnico di Torino
Torino, Italy

Luca Ardito
luca.ardito@polito.it
Politecnico di Torino
Torino, Italy

ABSTRACT

Gamification, the practice consisting of adapting game elements and features in non-recreational contexts to increase user motivation and interest, has become increasingly common in recent years in the different fields of Software Engineering such as development, requirements definition, testing, and education.

Among the different educational fields to which gamification has been applied, process modeling is currently not much explored: there are few examples of game-like approaches used for teaching process modeling, and such examples have yet to be applied for the duration of an entire course to assess possible benefits.

We thus describe the use of BIPMIN, a platform that implements elements regularly used in gamified tools such as levels, avatars, and leaderboards, in an Information Systems course, where students used the tool to perform practical BPMN modeling exercises over the whole duration of the course to get feedback on their modeling strategies.

The students' opinions have been gathered in the form of an end-of-course questionnaire and have been analyzed following the Straussian grounded theory approach to assess the general sentiment regarding usability, appreciation, and possible issues and improvement areas of the tool. The gathered results are encouraging, as they show that the tool has been well received and that its features that help student understanding the reasons behind their errors have been perceived as helpful for learning and improving BPMN modeling.

CCS CONCEPTS

• **Applied computing → Computer-assisted instruction**; **Interactive learning environments**.

KEYWORDS

Gamification, Process Modeling, Software Engineering Education, BPMN, Software Modeling

ACM Reference Format:

Giacomo Garaccione, Riccardo Coppola, and Luca Ardito. 2024. Gamifying Business Process Modeling Education: A Longitudinal Study. In *28th International Conference on Evaluation and Assessment in Software Engineering (EASE 2024), June 18–21, 2024, Salerno, Italy*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3661167.3661272

EASE 2024, June 18–21, 2024, Salerno, Italy

ACM ISBN 979-8-4007-1701-7/24/06
https://doi.org/10.1145/3661167.3661272

1 INTRODUCTION

Business Process Model and Notation (BPMN) is a graphical standard for representing business processes using a flow-charting logic for defining the high-level and low-level execution of a process; the notation can be used to specify the different actions that compose a process and their order, as well as the actors that execute each step and the way the actors interact [20]. BPMN is a notation that can be applied in Software Engineering, both in education and during requirements elicitation.

Gamification, the practice known as the use of elements commonly seen in games in non-recreational contexts to increase interest, participation, and motivation [7], has been adapted more and more in recent years to stimulate positive feedback and engagement, leading to increased productivity [13].

Despite its widespread adoption in different domains and fields of software engineering, gamification applied to process modeling remains a relatively unexplored field. Examples of game-like approaches to teaching process modeling, in particular, are rare and there is a lack of examples of it being applied to the context of an entire course.

We address this gap by using BIPMIN, a gamified tool for BPMN modeling, in an Information Systems course. The tool features established gamification mechanics such as levels, avatars, experience, and leaderboards to increase the motivation of students.

Our study is a prosecution of two previous articles [2, 11] , that defined a basic framework for gamifying process modeling education and first used this framework in a classroom environment. We expanded the mechanics defined in these two articles by reworking the evaluation engine, combining the separate leaderboard and progress mode into a single one, and expanding the feedback mechanism to be more detailed.

The novelty of our study comes from the fact that BIPMIN has been deployed for the entire duration of the course, allowing for a comprehensive assessment of its effectiveness in improving student understanding and performance in BPMN modeling exercises.

We evaluate the reception and effectiveness of BIPMIN among the students of the course through an end-of-course questionnaire that aims to gather insights regarding usability, appreciation, and possible issues and improvement areas; the answers are analyzed following the Straussian grounded theory approach.

Our research aims at contributing to the growing set of examples on gamification of software engineering education, by exploring a relatively new field. The findings of this study can offer insights into the potential benefits and challenges of adapting longitudinal gamification into a process modeling course.

The remainder of the paper is structured as follows: Section 2 contains a discussion regarding the state-of-the-art of BPMN modeling and on gamification being applied to software modeling in general, Section 3 describes the implementation of BIPMIN, and Section 4 describes the way we integrated it in the 2023-2024 of the course. We define the research questions that characterize this study in Section 5, discuss the results in Section 6 and possible threats to validity in Section 7; lastly, Section 8 presents the lessons learned from this study, the conclusions we draw and the future changes we intend to make to keep improving BIPMIN further.

2 BACKGROUND AND RELATED WORK

We present in this section an analysis of the current state-of-the-art gamification applied to software modeling, analyzing tools and platforms that cover both process modeling and other relevant topics such as conceptual modeling using Unified Modeling Language (UML) class diagrams.

2.1 Gamification

In recent years, there has been a significant rise in the application of gamification to the Software Engineering educational context [9, 23], leading to the definition of effective frameworks for gamification strategies.

An example of such a framework is Octalysis [5], defined by its creator as a *human-centred* design approach, which prioritizes understanding a user's emotions, feelings, and motivations instead of focusing on efficiency or results.

The framework defines eight *Core Drives*, specific human aspects activated by a gamified system that motivate users to perform and enjoy the offered activities. These eight drives are as follows:

- **Epic Meaning and Calling.** The feeling that one belongs to something greater and that their actions help reach a greater goal.
- **Accomplishment.** Being driven by achieving goals, progressing towards them, and learning new things.
- **Empowerment of Creativity.** The desire to express oneself, to have control of one's actions, and the act of receiving feedback after those actions.
- **Ownership.** The drive that increases interest as a consequence of giving people ownership and control over something.
- **Social Influence.** The human desire of wanting to interact and connect with other people, and to belong to a group.
- **Scarcity and Impatience.** The feeling of wanting something more and putting more effort into obtaining something rare or in high demand.
- **Unpredictability.** A drive that increases motivation by using novelty, surprises, and uncertainty.
- **Loss and Avoidance.** The feeling that drives users to act more carefully to avoid negative consequences and punishments.

The eight core drives are grouped according to two separate distinctions. One differentiates between extrinsic and intrinsic motivators. The former are linked to the left side of the brain and drive users to act by exploiting goals and rewards. The latter are associated with the right side of the brain and are rewarding by themselves, as they focus more on socialization and creativity, without a specific tangible reward.

The second distinction identifies White Hat drives (positive motivators that give a sense of control of one's action and satisfaction with their results) and Black Hat drives (negative motivators such as anxiety for the unknown or the risk of losing something).

According to the Octalysis framework, a gamified system is effective if it offers a balanced experience that satisfies both distinctions in all of its facets, meaning that a good system has both positive and negative motivators, as well as extrinsic and intrinsic at the same time. Focusing too much on only some aspects will lead to an unbalanced experience, and reduce the positive effects of gamification.

2.2 Gamification for Class Diagrams

Classutopia [19] is a mobile application that is used to teach good conceptual modeling practices in the form of a serious game. The game consists of a role-playing adventure where students face challenges in the form of defective diagrams that must be corrected, with a form of "combat" where errors and valid corrections damage the player and the enemy, respectively; visual feedback is also present, with errors being colored red and correct changes being green.

Júnior and Farias [14] present a framework for the gamification of software modeling education that focuses on the quality of the models. The mechanics used by the framework (points, progress indicators, feedback) are commonly seen in gamification theory and are applied to missions and scenarios where students are tasked with defining a specific class model.

The framework defines ten different quality metrics to assess a model such as syntactic quality (how much a model respects its language conventions), semantic quality (how well the model represents its domain), and scope quality (how the model is contextualized).

Cosentino et al. [6] describe a gamified plugin for Papyrus, a platform that is used for teaching modeling languages derived from UML such as class diagrams and use case diagrams. This plugin provides a challenge in the form of levels with increasing difficulty, with in-game rewards and achievements being given to the students who successfully solve the challenges, with a relevant focus on cheating prevention based on blocking information manipulation.

A similar plugin is Papygame [3], which makes use of different games such as Hangman to implement its modeling challenges with error feedback, awards, progression, and points as its game mechanics. Rules and game mechanics are implemented in a separate engine that allows for the definition of new types of games and exercises.

UMLegend [4] is presented as a prototype tool for teaching UML class diagrams with a focus on evaluating models depending on both syntactic and semantic quality; the prototype uses game mechanics such as levels, experience points, customizable avatars, and feedback to enhance the modeling activity. The authors mention that the evaluation system built into the tool identifies errors in a way that can be compared to how a human evaluator would.

2.3 Gamification for Process Modeling

Marín [18] describes a serious game (a game whose primary intended purpose is not enjoyment but learning [8]) that is used for teaching BPMN modeling by providing exercises where students have to complete diagrams by selecting the correct elements among the different available options.

The game offers challenging exercises of increasing difficulty with different tiers (basic, then medium once the basic tier is completed, then advanced when the medium tier is completed) and additional semantic questions to assess the student's understanding that contribute to an exercise's final score; achievements and leaderboards contribute to the challenge and reward the students' continuous efforts.

BPMS-Game [17] is a peculiar gamified tool that focuses on the sustainability of process modeling rather than on teaching good modeling practices. It makes use of common gamified elements such as achievements, awards, badges, and leaderboards, with rewards being given to models that follow the sustainability rules defined by the administrators; however, the models are analyzed according to said aspect exclusively, meaning that there is no focus on good modeling practices

Kutun et al. [15] present BPMN-Wheel, a competitive and cooperative board game that pits two teams against each other: players take turns spinning a wheel to obtain theoretical questions on BPMN modeling and, in case of correct answers, obtain in-game currency or process elements that can be used to build a target process; the winning team is the one that, at the end of the game, has produced the closer model to the intended reference solution. An experiment conducted using the game showed that students, after playing the game, improved the quality of their modeled processes.

3 THE BIPMIN TOOL

BIPMIN has been developed as a React-based web application that makes use of different open-source libraries to implement a BPMN modeler easily accessible to students. The tool is free to use and available online[1]. An empirical evaluation of the quantitative effects of the tool on students' performance has been performed on the 2022-2023 edition of the course [12].

3.1 Gamification Mechanics

BIPMIN offers multiple gamified elements in its current implementation, spread between mechanics that directly affect the execution of a student's exercise and mechanics that are applied separately, as a consequence of their actions. The mechanics are as follows:

- **Levels and Experience.** Levels act as a visual indicator of a student's modeling expertise, as they represent their progression in using the tool. Experience points are earned after completing exercises by reaching a high enough progress and can be multiplied depending on the performance (e.g. obtaining high percentages, completing exercises in few checks, and attempting exercises whose level is higher than the student's).
  The experience reward for each exercise is reduced in case the student makes errors and, to avoid making the experience too frustrating, it never goes below a minimum threshold; obtaining the minimum experience in every exercise is in any case enough to reach the maximum possible level.
- **Avatar Customization.** Avatars are used to give students a way to express their individuality thanks to a highly customizable implementation where students can change various details such as hair, clothing, facial hair, head decorations, and accessories. Students start with a few unlocked props and prop types and unlock more by leveling up. Avatar customization, with its unlockable props, is an example of the Ownership core drive.
- **Avatar Feedback.** The student's avatar appears on the exercise page and reacts accordingly to the actions performed by the student: more precisely, whenever checks made by the student contain errors that reduce the experience reward, the avatar will get sadder depending on the amount of lost experience.
  This implementation works as a negative motivator that leads students to be more careful in their activities to reduce the suffering of the avatar. An example of how this mechanic is used can be seen in Figure 1.1, where the avatar is represented with a sad expression connected to the low experience amount.
  An avatar's negative reactions to a student's mistakes implement the Loss and Avoidance core drive.
- **Leaderboards.** Leaderboards are used to implement a competition mechanism between students and offer a way for them to show their avatars and gauge their standing among their colleagues. Three different leaderboard categories rank the students depending on the number of exercises that have been completed, the level and experience points obtained, and the progress obtained for each exercise.
  Moreover, as a way to avoid a huge leaderboard listing all the enrolled students, the leaderboard screen shows two separate rankings: one leaderboard shows the top ten students for the corresponding metric, and another shows the relative position of the current students by showing the five closest students with higher scores and five closest with lower scores.
  Another important reason for having two separate leaderboards is that a complete leaderboard can be discouraging for students who start using the tool later and find themselves in the lower places of the ranking; having a relative leaderboard encourages students more, increasing their satisfaction when they can overcome their closest peers. Leaderboards implement the Accomplishment core drive.
- **Boss Icon.** An icon representing the *boss* of each exercise, a robot that challenges the student whenever they attempt the exercise. Completing the exercise by reaching the necessary process threshold defeats the boss and adds its icon to the corresponding exercise on the homepage.
  This mechanic ties the completion of an exercise with an in-game milestone, further increasing a student's satisfaction and increasing motivation for the subsequent exercises. The boss icon appears close to the student's avatar when attempting an exercise, as shown in Figure 1.2.

[1] https://softeng.polito.it/bipmin. Username: UserTest - Password: !!User_Test_s100030!!

The robot boss acts as a metaphor for the challenge a student is facing in each exercise and is an example of how to implement the Epic Meaning core drive.

- **Indicators.** Indicators are two separate progress bars shown on the exercise page that adapt to a student's diagram: the first bar represents the exercise's progress, that is, the number of correct elements, actors, and business entities present in the diagram, reduced depending on the presence of syntax errors and semantic errors such as elements being out of order or using names that are not allowed for the exercise. An exercise is considered completed by the tool if the progress reaches a specific threshold that can be customized for each exercise.
  The second bar, instead, displays the experience points that the student can earn once completing the exercise: this bar starts with the full amount of points available for each exercise and gets reduced in case the student makes mistakes. The experience bar acts as an indicator of the eventual reward, while the progress bar shows how close the student is to a correct solution of the exercise. Both bars are shown in Figure 1.3.
  The progress indicator is an example of the Accomplishment core drive while the experience indicator represents the Loss and Avoidance drive, as it gets reduced in case of mistakes.
- **Error Lists.** A set of lists that gets filled after a student's check in case there are some errors in the submitted diagram. Errors are divided into syntax errors, violations of BPMN modeling rules such as missing an end event in a process or not giving a name to a task, and semantic errors, violations that depend on the current exercise such as not having a necessary business entity, not having the necessary process parts, having the elements that compose a process part but in the wrong order, or using wrong message exchanges, for example.
  Each error in the list contains a textual explanation of the reason behind the error and a reference to the element that represents the error (where possible); additionally, different types of semantic errors are colored differently to better highlight the specific kind of mistake made, and how to address it. Error lists can be seen in the exercise page shown in Figure 1.4.
- **Process Parts Lists.** These lists follow the same reasoning as Error Lists, that is, explanations of the student's diagram after an evaluation, with the main difference being that they identify correct process parts, close process parts, and close elements.
  Correct process parts are sets of elements in the diagram that match those of the reference solution where each element is of the expected element type, has a name that is close enough to one of the expected ones, and belongs to the correct pool. Close process parts are process parts where at least half of the expected elements have a matching element in the diagram without all expected elements being present.
  Close elements are those elements that have a name that matches one expected name for an element in a process part but are of the wrong type, inside the wrong pool, or both.
  Each list directly references the elements found and, as shown in Figure 1.5, different lists are represented in different ways to reinforce the difference between correct, wrong, and close elements of a diagram.
- **Diagram Feedback.** A feature that ties directly to the lists described above, Diagram Feedback applies visual changes to a student's diagram: these changes include applying a green icon to an element that belongs to a correct process part, applying a purple shuffle icon to elements of a correct process part that are in the wrong order, applying yellow icons to close elements or of close process parts, or changing the border color of an element that belongs to the wrong lane or has an incorrect name, for example. The diagram represented in Figure 1 shows some examples of icons and color changes applied directly to a diagram.
  Diagram feedback and the two types of lists all represent the Empowerment of Creativity core drive, as they implement direct feedback as a consequence of a user's actions.
- **Exercise Hints.** The final gamified feature of BIPMIN consists of a set of unlockable hints that the students can access directly inside the exercise menu that also shows the text of the exercise (Figure 1.7).
  Hints can be unlocked after performing a certain number of checks, after reaching a high enough progress, or after losing enough experience; when defining an exercise, teachers can define multiple hints with the corresponding threshold, as long as they adhere to the three unlocking criteria.
  Hints are a representation of the Accomplishment core drive, being something that is unlocked as a consequence of a student's actions.

3.2 Evaluation Engine

One of the main features of BIPMIN, the ability to evaluate a student's diagram and provide direct feedback, depends on the evaluation engine that has been implemented in the tool.

Any time a student wants to have their diagram evaluated the engine performs two separate evaluations on the syntactic and semantic quality of the provided model, with the former being implemented using the external library *bpmn-js-bpmnlint*[2], that analyzes a diagram to identify syntactic violations such as having elements without a name, processes without either a start or end event or additional rules implemented directly in the tool that handle the way messages are exchanged by ensuring that only specific elements (e.g. message events, service tasks) can send messages.

The most relevant evaluation, however, is the semantic one, which works by allowing teachers to define a set of reference solutions for each exercise: a diagram is compared to all solutions, and the one most similar is taken as the reference, with its feedback given to the student. A reference solution defines the following entities:

- **Business Entities.** The pools that must be present in the diagram. A pool in the diagram matches a business entity if its name is close enough to at least one of the defined names for the entity; entities can also be defined as optional, in which case their absence is not considered an error, and

[2] https://github.com/bpmn-io/bpmnlint (last accessed on 21/02/2024)

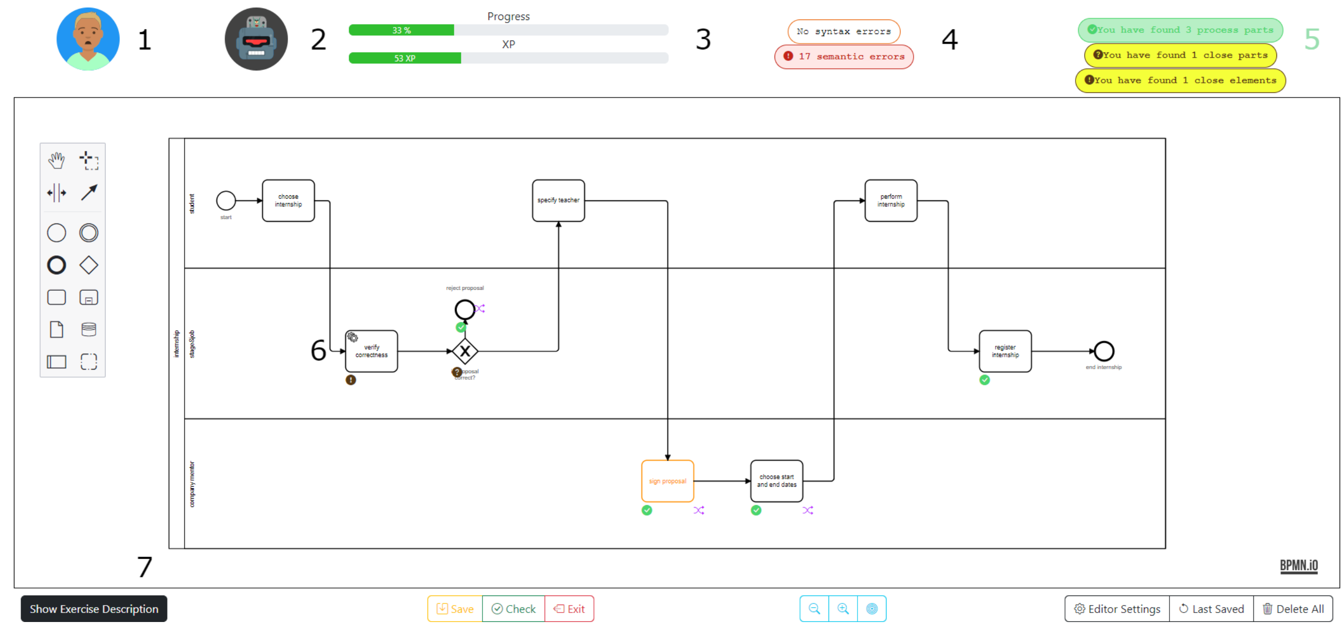


Figure 1: Exercise page of the tool showing: 1) Student Avatar, 2) Boss, 3) Indicator, 4) Error Lists, 5) Process Parts Lists, 6) Diagram Feedback, 7) Exercise Menu

as entities that must be represented using collapsed pools instead of expanded ones.

- **Actors.** The lanes that must be present in the diagram. A lane represents an actor if its name is close enough to at least one of the defined names and if it belongs to the specified business entity. In case the necessary business entity is not represented in the diagram, then its expected actors are automatically considered as missing.
- **Forbidden Entities and Actors.** Names that cannot be used for representing pools and lanes: examples may be using a name intended for a pool for a single lane, or representing the entire system as a lane of a process that is part of the system itself.
- **Sub-processes.** The sub-processes that must be modeled in the diagram. Each sub-process is characterized by the parent element (e.g. a pool or another sub-process) it needs to have, by whether it is an event sub-process or not, and, in case it is an event one if it also is an interrupting sub-process or not. An expected sub-process is missing if at least one of its constraints is not respected.
- **Process Parts.** The most important part of each reference solution, process parts represent the different logical parts into which a case study is divided. For each part, possible groups of elements can be defined as alternative ways to represent the part (e.g. a User Task followed by a Message Throw Event, or a User Task followed by a Service Task). Each element of each group has a set of allowed names, a set of possible element types it can be (e.g. a blank Task and a User Task can both be used to represent the same element), the pool and lane it must belong to, and additional information related to its relationships with other elements (e.g. preceding elements in the same group, elements with which it exchanges messages, eventual boundary events). An element of a group is modeled correctly if there is one element in the diagram that has one of the expected types, a name close to one of the expected ones, and belongs to the expected pool, all additional information is checked separately and is not required for a part to be present.

The evaluation engine works by performing these operations in the following order:

(1) Analyze the pools present in the diagram and find those that match the expected business entities by having a valid name; the entities without a matching pool are considered missing and are counted as an error each.
(2) Analyze the lanes present in the diagram and find those that match the expected actors by having a valid name and belonging to a pool that represents the expected business entity; the actors without a matching lane are considered missing and are counted as an error each.
(3) Analyze all pools that do not match an expected business entity and all lanes that do not match an expected actor: all pools and lanes with a name close to a forbidden one are counted as an error each.
(4) Analyze the sub-processes present in the diagram and find those that match the expected ones; all expected sub-processes without a match are considered missing and are counted as an error each.
(5) Repeat, for each expected process part, the following loop:
   (a) Repeat, for each possible group of elements, the following loop:

(i) Search among all the elements in the diagram for those that have a close name, the same type, and belong to the same pool as each expected element of the part.
(ii) If all the expected elements have a matching diagram element, then the group is found and the process part is modeled correctly. Otherwise, check if the group of matching elements is close enough (at least half of the expected elements have a matching diagram element and the group has more than the previous close group, with the first close group having 0 elements), and take the group as the new close group if that is the case.

(b) The process part is found if there is a matching group of elements, *close* if a close group has been found, and *missing* if there is no close group.

The *missing* and *close* process parts are counted as an error each.

(6) Analyze all the process parts that have been found to check additional errors (e.g. elements out of order, process parts out of order, wrong message exchanging, elements in the wrong lane)

The evaluation engine repeats the process above for each reference solution defined by the teachers, supporting different alternative representations of the same process in terms of both the overall high-level structure and of low-level elements of a single representation.

A limitation of the currently implemented engine, however, is the fact that all the alternatives must be imagined and defined by the teacher: the different representations of a process, the different ways to express a process's parts, and all possible synonyms for each name of each component of each solution must all be taken into consideration and be defined with as many alternatives as possible.

4 COURSE SETUP AND BIPMIN INTEGRATION

We adapted BIPMIN in an Information Systems course held at Politecnico di Torino, where the course is part of the first-year curriculum of the Master's Degree in Engineering and Management.

The course is organized in theory lectures and practical laboratory hours where students are encouraged to solve exercises on the different course topics such as conceptual modeling, use case analysis, cost estimation, and process modeling using BPMN.

In this edition of the course, BIPMIN was used as the tool for solving the BPMN exercises of the laboratories, as well as for the analysis of the BPMN section of a complex case study, resulting in a total of five exercises implemented on the platform; this marked a difference in comparison to the previous edition of the course, where students used the same tool (the Signavio Academic environment[3]) for all different exercises.

After a brief theory lecture that acted as the introduction to BIPMIN, students were given complete freedom in how and when they could tackle the exercises: the only specific deadline for the exercises was the end of the course, to allow students enrolled late to also try their hand at the tool.

A sufficient participation to the activities done with the tool was rewarded with two additional points added to the grade of the written exam: to qualify for these two points, students were asked to perform at least one attempt on each of the five exercises before the aforementioned deadline; moreover, we did not set a minimum threshold needed to qualify for the points, meaning that students would be rewarded even in case their attempts were not marked as *completed* by the tool, as long as their diagram showed at least the minimum effort. The course also involved a group-based project work consisting of the analysis of a full case study.

5 RESEARCH QUESTIONS AND METHODOLOGY

We define the following Research Questions to frame the analysis of our longitudinal experiment:

- **RQ1:** What is the *usability* of BIPMIN according to its users?
- **RQ2:** Is the longitudinal application of a gamified tool *appreciated* by students?
- **RQ3:** What *issues and improvement areas* emerge from the students' opinions?

To assess the general student perception of BIPMIN, we defined an end-of-experience questionnaire that was administered to the students during the final week of the course: the full questionnaire is available in the online Appendix[4]. Out of all the 363 students enrolled in the course, a total of 248 used the tool and answered our questionnaire.

To answer RQ1, we made use of both the Technology Acceptance Model (TAM) [16] and the GAMEX questionnaires [10]. We used the former to gauge the students' perception of the tool in terms of its usability, focusing on metrics such as Attitude towards usage (ATU), Perceived ease of use (PE), Behavioral intention to use (BI), and Perceived usefulness (PU).

The GAMEX, instead, was used to evaluate the gamified experience based on six different constructs: Enjoyment, Absorption, Creative Thinking, Activation, Possible Negative Effects, and Dominance. For both questionnaires, answers were given in the form of Likert-scale values ranging from 1 (Strongly Disagree) to 5 (Strongly Agree).

To answer RQ2, we defined a set of questions related to each gamified mechanic present in BIPMIN: for each mechanic, students were asked to leave their opinion on its usefulness, its influence on their usage of the tool, their appreciation of the mechanic, how well the mechanic motivated them in using the tool, and on whether the mechanic was seen as distracting.

Moreover, we included a set of questions focused on the perceived benefits of the longitudinal usage of the tool for the whole duration of the course. Answers to these sets of questions were, in the same way as those to the TAM and GAMEX questionnaires, expressed in the form of a Likert scale ranging from 1 to 5.

To answer RQ3, we included an open question that asked students to express their opinion on how BIPMIN could be better integrated into the course activities, as well as to express any issue they encountered when using the tool itself.

We analyzed the answers given by students by following the Straussian Grounded Theory approach [22], more specifically, the *open coding* approach, which let us define different topics present

[3]https://signavio.com (last accessed on 21/02/2024)

[4]https://doi.org/10.6084/m9.figshare.25259917.v1

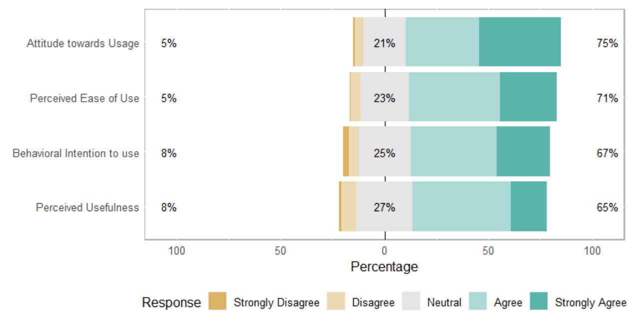


Figure 2: Distribution of answers to the TAM questionnaire

in the answers and then cluster the answers depending on each topic. We describe our approach below, according to the suggestions by Stol et al. [21] for applying grounded theory to software engineering, by describing the process and the distribution of roles

In our open coding process, we analyzed each answer, searching for a topic among the already defined ones that were compatible with the content of the answer; if the answer was compatible with an already existing topic, then we assigned the topic to the answer, while if no topic was compatible we defined a new topic and assigned it to the answer. The set of topics was thus built incrementally as we kept analyzing the answers.

Once all answers had been assigned a topic, all answers were then reviewed a second time to assess whether, for each question, the previously assigned topic was still appropriate or another topic was more suitable; in this second step, no new topic was defined, as the set of topics was considered finalized.

The open coding process described above was performed by one of the authors: the set of topics, and the assignment of topics to each question were then reviewed by all the authors together to reach a final consensus.

6 RESULTS

We present in this section the results of our analysis of the students' answers to the questionnaire. For all the questions whose answer was given in a Likert scale format, we considered, when computing the distribution of answers, the mean value of the answers given by each student for each aspect (e.g. the value of each student for *Attitude towards usage* was computed as the mean value of the four answers given by said student to the four questions related to the construct. For replication purposes, we provide the full set of answers as an online Appendix [5].

6.1 RQ1: Usability

Figure 2 shows the distribution of answers given to the TAM questionnaire.

We observe an encouragingly positive distribution of answers given to the TAM questionnaire: for all four fields, there are at least 65% of the students who either *Agree* or *Strongly Agree* as their general opinion regarding each field. The low percentage of students who *Strongly Disagree* for the Perceived ease of use and the Perceived usefulness are particularly relevant, as it means that BIPMIN is generally perceived as a tool that can be easily picked up and that can yield positive results.

Figure 3 shows a comparison between the answers given to the GAMEX questionnaire in two separate editions of the course: Figure 3(a) shows the answers given in the 2023 edition of the course, where BIPMIN was used during a single lecture and students only solved two exercises, while Figure 3(b) displays the answers relative to the 2023-2024 edition of the course.

The most obvious result of the comparison is a general improvement in the distribution of answers: while in the previous edition, positive answers ranged between 18% and 30%, the current implementation reaches far higher distributions, especially for *Enjoyment* and *Creative Thinking*, whose average scores greatly improve, making the current version more enjoyable from the students' points of view. Lastly, the *Possible negative effects* remains the question group with the highest number of negative answers, confirming that using BIPMIN is perceived as a positive and not frustrating experience.

Answer to RQ1: The answers to the TAM questionnaire show that BIPMIN is perceived as a high-usability gamified tool that is simple to use and effective for its intended purpose. The answers to the GAMEX questionnaire show that students perceived the gamified experience as enjoyable, and lacking in negative effects.

6.2 RQ2: Student Appreciation

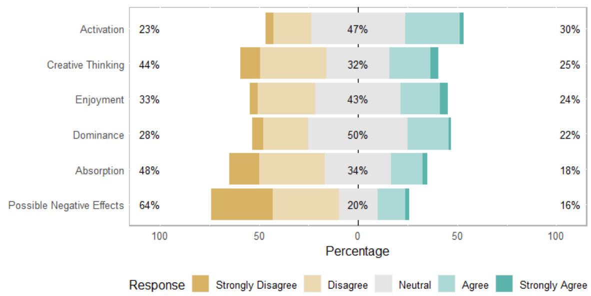
The distribution of answers related to the different game mechanics can be seen in Figure 4. When computing the answers related to each game mechanic, we considered the opposite value regarding the negative question of each mechanic (e.g. if the answer given to the question *Indicators distracted me during my usage of BIPMIN* was *Disagree* the value used for calculating the mean was *Agree*, and so on).

The distribution of answers shows that most of the game mechanics have been fairly appreciated by the students of the course: particularly worth mentioning is the fact that the four most appreciated ones (Diagram Feedback, Indicators, Error Lists, and Process Parts Lists) are all mechanics that directly impact a student's understanding of their diagram. These four mechanics are those that assist a student by pointing out what are the correct elements inside a diagram, showing how close the diagram is to the reference solution, listing the errors and missing elements, as well as listing the correct parts; the fact that at least 60% of the students considers these mechanics as beneficial is an encouraging positive result, as it highlights one of the greatest strengths of BIPMIN, one that will need to be expanded further to improve the tool even more.

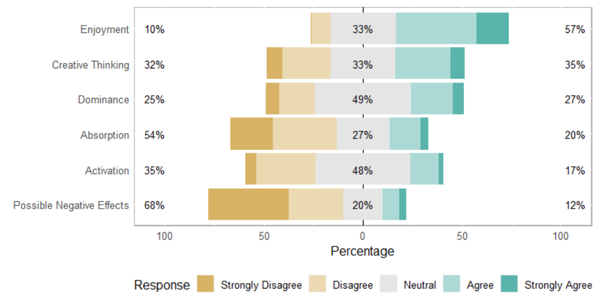
Figure 5 shows the distribution of answers related to the longitudinal usage of the tool. The distribution of answers regarding the longitudinal application of the tool has been computed for each single question, without computing any mean value.

The answers related to the longitudinal application of BIPMIN show a positive sentiment shared by most of the students, who believe they have learned the basic rules of BPMN modeling and

[5] https://doi.org/10.6084/m9.figshare.25259917.v1



(a) Distribution of answers to the GAMEX questionnaire in the 2023 application of BIPMIN



(b) Distribution of answers to the GAMEX questionnaire in the 2024 application of BIPMIN

Figure 3: Comparison between the answers given to the GAMEX questionnaire for the previous edition of the course and the 2023-2024 edition

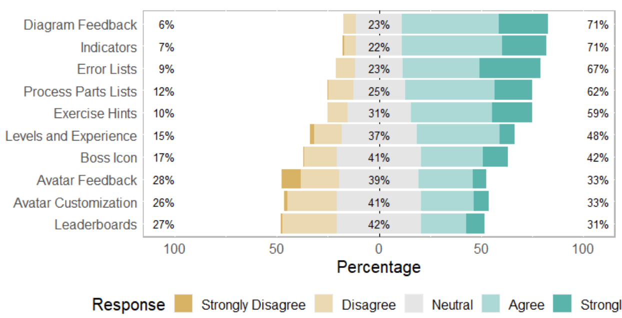


Figure 4: Distribution of answers related to the different game mechanics

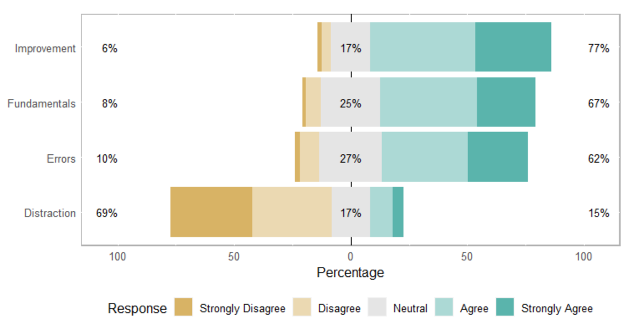


Figure 5: Distribution of answers related to the longitudinal usage of BIPMIN

that, by continuously using the tool, they have managed to improve their knowledge of modeling. It is also worth mentioning that the majority of the students do not believe that using BIPMIN was distracting, as they feel they managed to focus on the other course activities.

Answer to RQ2: The answers related to the different game mechanics show that students have appreciated the way gamification was implemented. The most relevant finding of these questions is the fact that students greatly appreciate the mechanics that directly help them understand what is correct and what is missing. The longitudinal usage of the tool was also appreciated, as most students believe that using the tool was helpful for both learning the fundamentals of modeling as well as improving their general knowledge.

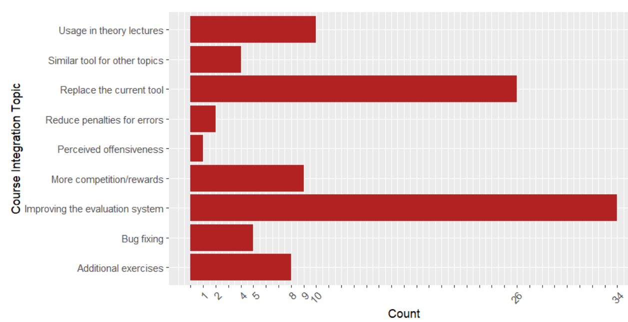


Figure 6: Topics found in the open-ended question after performing open coding on the answers

6.3 RQ3: Issues and Improvement Areas

We present in Figure 6 the results of the open coding we performed on the answers given to the open-ended question asking for the students' opinions. Out of the 248 total answers, we identified 99 answers that did not contain any opinion, issue, or suggestion, and 50 answers where students declared themselves satisfied with the way BIPMIN was used in the 2023-2024 course; these answers have not been counted in the analysis, leading to a total of 99 answers considered.

The most common topic found in the open-ended questions (one-third of the total analyzed answers) highlights the current limits of the evaluation system: a student's diagram is evaluated in comparison to a set of reference solutions, an implementation strategy that requires teachers to consider all possible alternative modeling solutions for a given exercise, in terms of both diagram elements and possible names given to the different elements; many students found themselves in a situation where a diagram that was assigned a low score was judged much higher after minor changes such as using a synonym for an element's name or changing some elements in the diagram.

Other relevant topics emerged as a consequence of using BIPMIN exclusively in the practical laboratories: theory lectures were still conducted using Signavio Academic, and the same platform was the one students had to use for the final project work. This disparity in usage, as well as the absence of gamified tools for the other course topics, led many students to ask for a more thorough integration of gamification, spanning over the entire course instead of just the laboratories.

Lastly, it is important to mention that one student perceived the way avatars were implemented as stereotyped and offensive: the student did not appreciate the randomly generated initial avatar and commented that *the effort to create more representation was just a very generalized approach which made me feel very stereotyped as an arab woman who doesn't come from a Muslim nor African background*; in the student's opinion, using a blank avatar would have been more beneficial and less offensive.

Future development will need to assess this issue very carefully, to make sure that the gamification features are not seen as discriminatory, offensive, or stereotyped; either a default state for the avatar or a different implementation is going to be necessary to make the experience as inclusive as possible.

Answer to RQ3: The students' opinions showed that the evaluation system needs to be improved by considering the many possible alternative ways in which a concept can be expressed, both in textual form as well as in terms of different diagram elements. The current integration of BIPMIN in the course is also perceived as somewhat lacking, as many different activities have been performed without gamification, leading to a dissonant situation where students feel a desire to have gamification be used for all possible activities. Last but not least, the implementation of the avatar system can risk to be perceived as stereotyped and offensive, and corrective actions are going to be necessary going forward to make the tool more inclusive.

7 THREATS TO VALIDITY

We discuss in this section the possible biases that may have affected the execution of the study, and may thus undermine the results and the conclusions we draw from them.

A possible bias comes from the fact that students who used the tool would be awarded two extra points for the written exam: this promise may have attracted only the more motivated and interested students, thus meaning that the tool may not be received in the same way by less interested students.

Another possible bias is related to the fact that students may be more inclined to give positive answers to the final questionnaire rather than be truthful: a different set of students may give different answers, meaning that we cannot consider our conclusions as final.

Lastly, there may be a difference between the experience and ability of students and the answers they gave to the questionnaire: more skilled students may tend to give more positive answers, while students who underperform may be dissatisfied and thus tend to give unfavorable answers. This may introduce a correlation between students' final results and their answers, and we did not consider that to be the case.

8 CONCLUSION AND FUTURE WORK

This study described the longitudinal usage of a gamified tool for BPMN modeling in an Information Systems course and analyzed the students' opinions on the experience in terms of usability, appreciation, issues, and possible improvements.

The lessons we learned from conducting the experience are as follows:

- Continuous usage of a gamified tool improves the way the tool itself is perceived by students: compared to just using the tool once, having access to BIPMIN for the duration of the course made students appreciate it more and perceive it as more enjoyable, usable, and effective.
- The most appreciated mechanics are actually outside of those that are commonly seen in gamified tools: students preferred the mechanics that gave immediate and direct feedback related to a student's model. In turn, continuous and dedicated feedback led students to easily learn the basics of BPMN modeling and improve their knowledge the more they used the tool.
- The evaluation system implemented in a learning tool needs to be as flexible as possible and allow different alternative solutions for the same exercise: the most common issue identified by the students was that the evaluation was too dependent on using both the expected words and the expected sequences of elements for a diagram to be considered correct. Alternative solutions must be considered by teachers to facilitate the learning process.
- Using separate tools for different course topics, especially if they provide different strategies (e.g. one uses gamification while others do not), can lead to excessive student overload. A useful strategy is to implement all topics and activities inside a single tool: an example is the Signavio Academic platform, which is currently adopted for non-BPMN-related topics in our course and is an example to follow for the development of similar gamified tools for other topics.

The results of this study are encouraging and warrant additional research work on the topic, as we plan to perform an empirical evaluation that will focus on the commitment of the students in using BIPMIN in this edition of the course, as well as a comparison of the effectiveness of the tool regarding student grades between this edition of the course and the previous one.

Furthermore, we plan to address the issues that have emerged as a result of our analysis by implementing gamified tools for the other topics of the course such as UML class diagrams, use case diagrams, use case narratives, and cost estimation: we believe that using a single gamified tool for all the activities of a course should lead to improvements in both student interest and performance.

Lastly, we plan to improve the evaluation system of the tool by making it more flexible and able to recognize more synonyms and different semantical interpretations of a case study: a possible strategy worth exploring is the usage of Large Language Model agents that have shown to be particularly effective at this kind of task [1].

ACKNOWLEDGMENTS

This study was carried out within the "EndGame - Improving End-to-End Testing of Web and Mobile Apps through Gamification" project (2022PCCMLF) – funded by European Union – Next Generation EU within the PRIN 2022 program (D.D.104 - 02/02/2022 Ministero dell'Università e della Ricerca). This manuscript reflects only the authors' views and opinions and the Ministry cannot be considered responsible for them.

REFERENCES

[1] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. Program Synthesis with Large Language Models. _eprint: 2108.07732.

[2] Kylie Bedwell, Giacomo Garaccione, Riccardo Coppola, Luca Ardito, and Maurizio Morisio. 2023. BIPMIN: A Gamified Framework for Process Modeling Education. *Information* 14, 1 (2023). https://doi.org/10.3390/info14010003

[3] Antonio Bucchiarone, Maxime Savary-Leblanc, Xavier Le Pallec, Antonio Cicchetti, Sébastien Gérard, Simone Bassanelli, Federica Gini, and Annapaola Marconi. 2023. Gamifying model-based engineering: the PapyGame experience. *Software and Systems Modeling* (2023), 1–21. Publisher: Springer.

[4] Christian Cagnazzo, Giacomo Garaccione, Riccardo Coppola, Luca Ardito, and Marco Torchiano. 2023. UMLegend: A Gamified Learning Tool for Conceptual Modeling with UML Class Diagrams. In *Proceedings of the 2nd International Workshop on Gamification in Software Development, Verification, and Validation (Gamify 2023)*. Association for Computing Machinery, New York, NY, USA, 2–5. https://doi.org/10.1145/3617553.3617883

[5] Y. k Chou. 2015. *Actionable Gamification: Beyond Points, Badges, and Leaderboards*. Createspace Independent Publishing Platform. https://books.google.it/books?id=jFWQrgEACAAJ

[6] Valerio Cosentino, Sébastien Gérard, and Jordi Cabot. 2017. A Model-based Approach to Gamify the Learning of Modeling.. In *SCME-iStarT@ ER*. 15–24.

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

[8] Damien Djaouti, Julian Alvarez, and Jean-Pierre Jessel. 2011. Classifying Serious Games: the G/P/S model. *Handbook of Research on Improving Learning and Motivation through Educational Games: Multidisciplinary Approaches* (Jan. 2011). https://doi.org/10.4018/978-1-60960-495-0.ch006

[9] Daniel J. Dubois and Giordano Tamburrelli. 2013. Understanding Gamification Mechanisms for Software Development. In *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering* (Saint Petersburg, Russia) *(ESEC/FSE 2013)*. Association for Computing Machinery, New York, NY, USA, 659–662. https://doi.org/10.1145/2491411.2494589

[10] René Eppmann, Magdalena Bekk, and Kristina Klein. 2018. Gameful experience in gamification: Construction and validation of a gameful experience scale [GAMEX]. *Journal of interactive marketing* 43, 1 (2018), 98–115. Publisher: SAGE Publications Sage CA: Los Angeles, CA.

[11] Giacomo Garaccione, Riccardo Coppola, Luca Ardito, and Marco Torchiano. 2023. Gamification of Business Process Modeling Notation education: an experience report. In *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering (EASE '23)*. Association for Computing Machinery, New York, NY, USA, 460–464. https://doi.org/10.1145/3593434.3593956 event-place: Oulu, Finland.

[12] Giacomo Garaccione, Riccardo Coppola, Luca Ardito, and Marco Torchiano. 2024. Gamification of business process modeling education: an experimental analysis. *Software and Systems Modeling* (2024), 1–26.

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

[14] Ed Júnior and Kleinner Farias. 2021. ModelGame: A Quality Model for Gamified Software Modeling Learning. In *Proceedings of the 15th Brazilian Symposium on Software Components, Architectures, and Reuse (SBCARS '21)*. Association for Computing Machinery, New York, NY, USA, 100–109. https://doi.org/10.1145/3483899.3483910 event-place: Joinville, Brazil.

[15] Bahar Kutun and Werner Schmidt. 2019. BPMN wheel: Board game for business process modelling. In *European conference on games based learning*. Academic Conferences International Limited, 1008–1012.

[16] Younghwa Lee, Kenneth A Kozar, and Kai RT Larsen. 2003. The technology acceptance model: Past, present, and future. *Communications of the Association for information systems* 12, 1 (2003), 50.

[17] Javier Mancebo, Felix Garcia, Oscar Pedreira, and Maria Angeles Moraga. 2017. BPMS-Game: tool for business process gamification. In *Business Process Management Forum: BPM Forum 2017, Barcelona, Spain, September 10-15, 2017, Proceedings 15*. Springer, 127–140.

[18] Beatriz Marín. 2022. Lessons Learned About Gamification in Software Engineering Education. 1473–1496. https://doi.org/10.4018/978-1-6684-3710-0.ch071

[19] Beatriz Marín, Felipe Larenas, and Giovanni Giachetti. 2018. Learning Conceptual Modeling Design Through the Classutopia Serious Game. *International Journal of Software Engineering and Knowledge Engineering* 28 (Nov. 2018), 1679–1699. https://doi.org/10.1142/S0218194018400235

[20] OMG. 2011. *Business Process Model and Notation (BPMN), Version 2.0*. Needham, MA, USA.

[21] Klaas-Jan Stol, Paul Ralph, and Brian Fitzgerald. 2016. Grounded Theory in Software Engineering Research: A Critical Review and Guidelines. In *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*. 120–131. https://doi.org/10.1145/2884781.2884833

[22] Mai Thai, Li Chong, and Narendra Agrawal. 2011. Straussian Grounded Theory Method: An Illustration. *Qualitative Report* 17 (Sept. 2011). https://doi.org/10.46743/2160-3715/2012.1758

[23] Vladimir Uskov and Bhuvana Sekar. 2014. Gamification of software engineering curriculum. In *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*. 1–8. https://doi.org/10.1109/FIE.2014.7044098