A standard framework for gamification evaluation in education and training of software engineering: an evaluation from a proof of concept

Rodrigo Henrique Barbosa Monteiro
Graduate Program in Computer Science (PPGCC)
Federal University of Pará (UFPA)
Belém, Pará, Brazil
rodrigo.monteiro@icen.ufpa.com

Sandro Ronaldo Bezerra Oliveira
Graduate Program in Computer Science (PPGCC)
Federal University of Pará (UFPA)
Belém, Pará, Brazil
srbo@ufpa.br

Maurício Ronny de Almeida Souza
Graduate Program in Computer Science (PPGCC)
Federal University of Lavras (UFLA)
Lavras, Minas Gerais, Brazil
mauricio.ronny@ufla.br

***Abstract*—This Research to Practice Full Paper presents that the gamification has been used to motivate and engage participants in software engineering education and training. The effects of gamification in this area have been studied and reported since 2011. However, there are no studies that propose standard procedures for evaluating gamification in the specific context of software engineering education and training. As a result, each study proposes their own evaluation procedures, making it difficult to compare approaches. The standardization of criteria, measures and indicators can allow an objective comparison between primary studies on gamification in software engineering, reinforcing results and revealing trends. Thus, the objective of this study is to propose and evaluate a framework for the evaluation of gamification in the context of software engineering education and training. The proposed framework focus on the structural aspect of an evaluation, consisting of concepts (entities) and their relationships. The development of this framework consisted of three steps: (1) the definition of the framework structure, based on the results of a systematic review of the literature on evaluation of gamification in software engineering education and practice, and its adequation to the GQIM (Goal-Question-Indicator-Metric) model; (2) an ad hoc review of this framework by three researchers ; and (3) the execution of a PoC (Proof of Concept) evaluation, in order to perform an preliminar assessment of the framework adequacy. As a result, we were able to use the framework to model the evaluation of two primary studies, documenting the items found in these studies, which revealed the existence of a common measure between the two studies (Total lines of code, or LOC). The existence of only one common measure in both evaluations makes it difficult to compare the results of the studies. Additionally, we observed that one of the studies did not present information to justify the choice of the measures used in its evaluation and did not summarize all the data collected in its evaluation procedures. Therefore, the use of the framework might be useful to identify points of improvement for the validity of evaluation studies.**

Keywords— gamification, software engineering, proof of concept, evaluation framework, training, education.

I. Introduction

The use of gamification as a motivational tool in education and training has been the subject of studies for the past 10 years [1] [2]. However, there is still a lack of empirical data to evaluate the impacts of gamification, either due to the novelty of gamification literature [2], the lack of details on evaluation design in experiments or quasi-experiments [3], for the absence of significant statistical data or for reporting inaccurate qualitative data from students [4].

This study presents an structural framework for the evaluation of gamification in software engineering. The goal is to provide a standard structure for researchers and gamification designers in order to plan evaluation studies. The components of the framework are derived from results of previous study [1], which mapped evaluation strategies,methods and criteria in gamification literature in the context of software engineering education and practice. The framework was evaluated by means of PoC (Proof of Concept), applying it in two primary studies on gamification evaluation. In our results, we were able to instantiate the framework in both primary studies, documenting the items found in those works.

As a consequence of using the framework, we were able to identify improvement opportunities in one of the studies regarding the lack of information that justifies the choice of the metrics used in its evaluation and lack of details on the collected data. We believe that, in addition to supporting the design of evaluation studies, our framework may also encourage standardization of evaluation studies that may allow verification and comparison of different gamification approaches.

In addition to this introductory section, this paper is structured as follows: Section II presents the theoretical basis for understanding the research carried out, Section III discusses some related works, Section IV presents the evaluation framework for gamification, Section V presents the preliminary evaluation to verify the adequacy of this framework, Section VI discusses some threats to validity and, finally, Section VII presents the conclusions of this paper.

II. Background

Gamification is the application of game mechanisms in non-game contexts [5]. It is a playful approach that contributes to increasing people's motivation and engagement, if properly applied to a context.

This approach has been studied since 2011 and the pace of publications of empirical research on the subject has been increasing ever since [2] [1]. However, more empirical data is needed to support its beneficial effects [4] [3] [2] [1].

In a previous study [1], we mapped 100 primary studies (from 2011 to 2020) on the state-of-the-art and the state-of-practice of gamification in software engineering and analyzed the methods used in the evaluation of gamification. Sixty four (64) studies report evaluation procedures, and only three studies [6]–[8] propose models for the evaluation of gamification, yet there are no evidences of use of these models. The results of the study identified that evaluation of gamification in software engineering focus on two aspects: the evaluation of the gamification strategy itself, related to the user experience and perceptions; and the evaluation of the outcomes and effects of gamification on its users and context. There is no general consensus on the criteria, metrics and indicators that are used in the evaluations, due to the quantity and diversity in which they are present in these studies. However, the study shows the most recurring criteria for the evaluation are "engagement", "motivation", "satisfaction", and "performance" [1]. Finally, the authors highlight that the evaluation of gamification requires a mix of subjective and objective inputs, and qualitative and quantitative data analysis approaches [1].

Therefore, it is clear that each study on gamification proposes its own evaluative approach and identifies the main aspects to be analyzed within its specific scope. This phenomenon makes difficult to consolidate the few empirical data available, because: these projects may take time to mature and provide significant empirical data on the use of gamification and the methodological diversity of the evaluation makes it impossible to compare the empirical data provided by these studies.

As a result, the development of an evaluation framework for gamification in software engineering, which considers the state-of-the-art and the state-of-practice of gamification, allows new studies to plan the evaluation considering what has been commonly applied until then, thus contributing to: (i) the standardization of generic entities present in the design of the evaluation of gamification in research already carried out in the context of gamification in software engineering; (ii) streamlining the design process of the evaluative approach in future research; (iii) the comparison of different gamification surveys where the framework is instantiated; and (iv) assistance in generating significant empirical data for understanding the impact of gamification in different contexts, situations and users.

III. Related Works

The current paper uses the results of a previous study [1] (described in Section II) as the basis for the conception of a framework for the evaluation of gamification in software engineering. To the best of our search, there are few studies describing gamification frameworks in software engineering, that focus or contemplates the evaluation stage [6] [7] [8]. However we did not find evidences or reports of their use in the context of teaching, training or industry of software engineering. Therefore, our framework considers insights from these studies in its design, and in this paper we describe an evaluation study using PoC method.

In the context of serious games and Game-based Learning in software engineering education, Petri et al. [9] presents MEEGA+, a framework for the evaluation of educational games in software engineering. To analyze the reliability and validity of their framework, the authors carried out 48 case studies, with 843 participants, and evaluated the results obtained through the evaluation applied in these studies, and concluded that MEEGA + is reliable. In this study, the authors adopted the GQM (Goal-Question-Metrics) approach to define evaluation steps and to guide the instance of entities to be instantiated (objectives, questions and metrics) [10]. There is also the GQIM (Goal-Question-Indicator-Metrics) approach, which is an extension of GQM. The GQIM is used to identify metrics and indicators consistent with the objectives and questions of an assessment in Software Engineering [11]. To analyze the reliability and validity of their framework, the authors carried out 48 case studies, with 843 participants and concluded that MEEGA + is reliable. We believe that frameworks such as MEEGA+ are important for the consolidation of empirical data in a given knowledge area. Therefore, our framework is inspired by MEEGA+. However, the current state of our framework focus on the structural aspect of evaluation studies in gamification.

Regarding the use of PoC as an evaluation strategy, some primary studies on approaches related to games [12] and on gamification [13] [14] applied this method to evaluate the viability of their models. In these studies, there was the design of a structure of procedures and classes, which were implemented through programming and analyzed with instances within prototypes.

IV. Evaluation Framework for Gamification in Software Engineering

This section presents the conception, elaboration and construction of the evaluation framework for gamification in software engineering.

A. Conception Methodology

The framework design process was planned and executed considering the findings on the evaluation of gamification in software engineering in our previous study [1], and the GQIM model (Goal-Question-Indicator-Metric) used for the development of the MEEGA + framework [9]. We also considered the PoC evaluation method used to analyze the instances of Sripada, Reddy and Khandelwal game-related frameworks [13] and Parizi [14]. Figure 1 illustrates the design steps, which include PoC evaluation (details in Section V).

Three researchers participated in the data synthesis, the ad hoc review and the PoC: a graduate student in computer science and two PhD professors / researchers in software engineering. At the end of each design step, the researchers held a committee to discuss the results, to propose refinements, and and to decide to proceed to the next stage. Altogether, this study lasted two months, starting in December / 2020 and ending in March / 2021.

Section IV-C presents the mapping of the evaluation phases found in the findings of the systematic mapping [1]. Section IV-D explains the framework, composed of entities and relationships, modeled after the GQIM and the findings of the systematic mapping.

The PoC will be better described in Section V Despite this, it is fundamental for the framework design. During the process of establishing entities on each primary study, new information can be included, reviewed, or excluded from the framework design. Thus, the PoC step performs both the evaluation function and the review function [15].

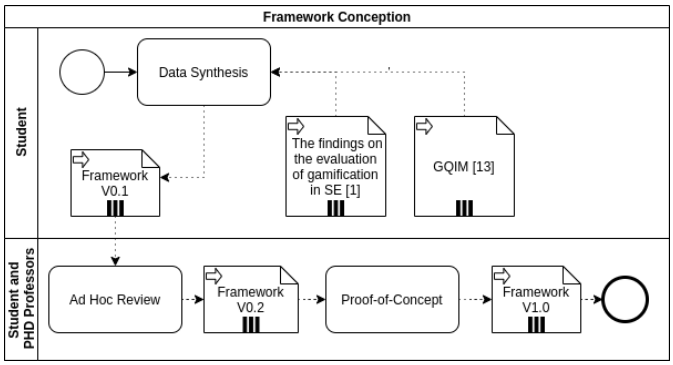

Fig. 1. Sequence of activities and work products of the design the evaluation framework.

TABLE I
MAPPING OF COMMON PHASES BETWEEN THE EVALUATION MODELS FOUND

Generic Phase	Phase in Primary Study
Contextualization of Gamification	Precondition [6] [8], Basic Component [7]
Contextualization of Evaluation	Precondition [6] [8]
Definition of Methods	Target [8], Procedural Component [7], Metrics Configuration [8]
Summary of Results	Obtained Results [8], Results Component [7]
Analysis of Results	Results Component [7], Evaluation Success [8]

B. Instrumentation

In the conception step, instruments of analysis, modeling and remote communication were used in the activities of Data Synthesis and Ad Hoc Review. Data Synthesis Instruments: Google Sheets (Electronic Spreadsheet) for the analysis of the summary data obtained in [1], Google Docs (Text Editing) for the preparation of analysis reports used in the Ad Hoc Review, Draw IO (Diagramming) tool for the elaboration of activity diagrams (BPMN - Business Process Modeling Notation) and entity-relationship diagrams. Instruments of the Ad Hoc Review: Google Meet (remote meeting) to carry out discussions on the framework design between the three researchers.

C. Evaluation Phases

In our previous study [1], we found three evaluation models of gamification in software engineering [6]–[8]. Each of the studies defined evaluation phases. It was possible to identify generic phases within these models from a qualitative analysis of their studies. Table I presents the mapping of the generic evaluation phases found in the three studies and the phases that belong to their models as stated by the authors. After that, each of the phases is explained.

- Contextualization of Gamification: Definition of the gamified approach (with its dynamics, rules and emotions) of the context in which the approach is applied,
- Contextualization of Evaluation: Definition of the actors of the gamified approach (participants) and the context in which the evaluation is carried out,
- Definition of Methods: Definition of data collection methods,
- Summary of Results: Data collection and extraction of information to be analyzed,
- Analysis of Results: Analysis of the summarized information.

Some generic phases are not present in all models, as is the case of the Contextualization of Gamification, Contextualization of Evaluation and Analysis of Results phases. However, the contextualization is not presented as an evaluation procedure in these studies, but as a necessary precondition. The same is not true in [6], as the results analysis is not really presented in this study.

D. Generic Entities

A evaluation framework for gamification in software engineering is a structure composed of rules and ideas, which allow the evaluation planning of a gamified approach. Considering that the information pertinent to the evaluation of gamification is related in different evaluation phases, the proposed framework is constituted of entities and relationships, which can be instantiated in different evaluation phases.

Figure 2 shows an entities-relationships diagram using UML (Unified Modeling Language) notation for an overview of the information that needs to be obtained for the gamification evaluation report. The explanation of each of these entities also follows.

- Contextualization of Gamification:
  - Gamification: contains the main information that describes the gamified approach to be analyzed and the context in which the gamified approach is applied,
- Contextualization of Evaluation:
  - Evaluation: contains the main information about the evaluation of the applied gamified approach. All other entities are derived from it, as each evaluation is unique when its scope is delimited by empirical research and all knowledge obtained is validated from it,
  - Objective: like any evaluation, scientific research has well-defined objectives. The definition of objectives is not only present in the study-of-practice of gamification, but it is also fundamental for the evaluation design, as it delimits the discussion to solve a problem. Depending on the purpose of the evaluation, this entity is subdivided into criteria,
  - Criteria: while the same evaluation has one or more general objectives, the evaluators can break these objectives down into a set of criteria, in order to divide the problem into subproblems, which are easier to analyze and delimit issues,
  - Questions: these are objective questions, in order to be answered by the results of the evaluation, which confirm (or not) the predicted hypothesis. These questions justify the choice of metrics, indicators and evaluation instruments,
- Definition of Methods:
  - Measures: determine the nature of the data to be collected,
  - Indicators: define the information expected from the data collection that corroborates the predicted hypothesis,
  - Instruments: are the means to be used for data collection, depending on the measurement used,
- Summary of Results:
  - Rounds: is the set of periods when data collection and analysis must take place. Each round can have information that details the sample to be used as a parameter for collection and analysis,
  - Sample: details the population that participates in the evaluation in a specific round. It must be specified whether a sample was submitted to a gamified approach, or not, when more than one round is applied, with at least one of the rounds being composed only of control samples, and another one, only of experimental samples,
  - Collection: represents the data collected at the time the round is being executed,
  - Sample Collection: represents the data collected at the time the round is being performed, considering an individual sample,
- Analysis of Results:
  - Analysis: set of information that is obtained from the study (qualitative or quantitative) of one or more collections, on one or more measures, within one or more evaluation rounds, in order to confirm the hypotheses raised in the questions to be answered,
  - Finding: represents the discussion about the confirmation of the hypotheses raised, questioning whether the questions were in fact answered, what consequences these responses generate, and whether there is a need for further evaluations on the subject. In short, it discusses the gamified approach applied based on the careful analysis of the collected data.

V. FRAMEWORK EVALUATION

This section presents the evaluation method and results of the evaluation framework for gamification in software engineering.

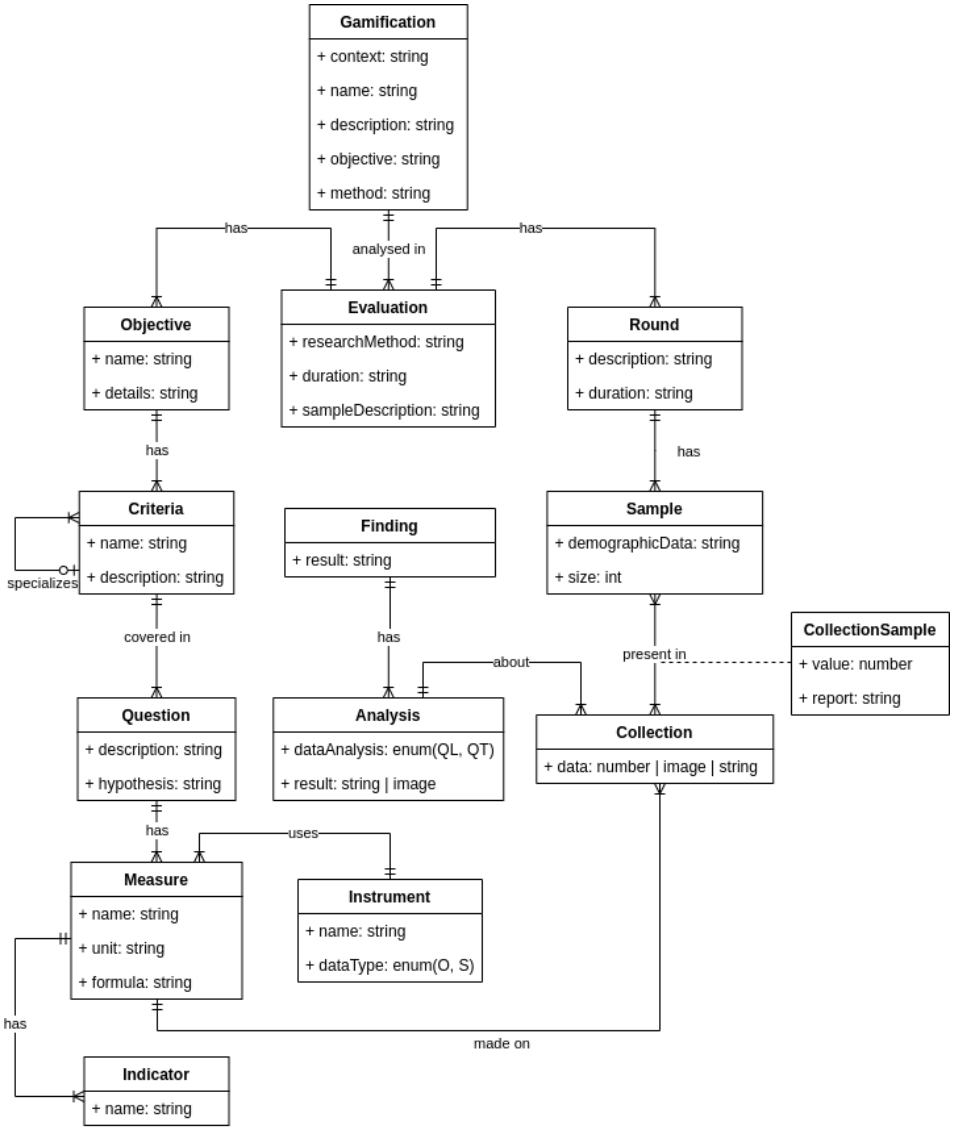

Fig. 2. Entity-relationship diagram of the evaluation framework for gamification in software engineering

A. Evaluation Methodology

The PoC does not have a single definition, as this terminology is used in many research areas, and there is no consensus on its definition and characterization [16] [15]. However, in software engineering, it can be understood as a technique used to verify and analyze the feasibility of materializing an idea [17].

We use the PoC process in order to evaluate the feasibility of using the evaluation framework for gamification in the software engineering.

Neto, Borges and Roque [15] analyzed the adoption of PoC in five information technology companies, documented ten common practices and characterized them based on the CE framework (Context Engineering). The CE framework presents a set of movements to be carried out depending on the need for an organization, demanded by the social interaction between the participants and by the interaction with work products of that organization.

Following this definition, we explored (“Exploring”) the state-of-the-art and the state-of-practice of gamification in software engineering, in order to understand (“Comprehending”) how this evaluation works, to then model (“Modeling”) the abstract entities that can be generalized; we specified (“Specifying”) what information could be extracted from the primary studies on gamification, in order to be instantiated (“Execution”); we deliberate (“Negotiating”) on how the framework items were instantiated, we apply improvements (“Improvising”) when we agree that there is a need, and we document the applied changes (“Documenting”).

The steps above were performed sequentially in two primary studies present in [1], always applying all modeled abstractions and modifying the framework when necessary, already using this new version in the PoC of the next study case.

B. Instrumentation

In the evaluation step, modeling and remote communication instruments were used in the PoC activity: Draw IO (Modeling) for the elaboration of object diagrams (using UML notation) referring to the instantiation of the information present in the PoC evaluations, and Google Meet (remote meeting) to conduct discussions on instantiations and revisions of the framework.

C. Results

This section presents the results of the PoC applied in these two studies. Data analysis consists of presenting the information used in the instance of the entities, the inadequacies found in each instantiation and the changes made to readjust the framework to the study previously analyzed.

1) First Primary Study [18]: The instantiation of the evaluation framework for this primary study was carried out as follows, considering the values defined for the attributes of each entity:

- **Gamification**: *Context* - software development; *Name* - SGM (Software Gamification Model); *Description* - this is a software gamification model (SGM) with a well-defined structure that provides a robust gamification implementation process for Software Engineering. This model also contains elements of applied psychology, Social Software Engineering (SSE) and Capacity Maturity Model (CMM). Each participant receives a badge for every 10 points acquired. For every 20 points or 2 badges, the participant receives a virtual gift. Points and badges are displayed on a leaderboard; *Objective* - improving software quality and collaboration in the context of cross-cultural software development; *Method* - a SharePoint community is used as a means of collaboration and points collection,
- **Evaluation**: *Research Method* - experiment; *Sample Description* - software engineers from the USA, Western Europe and Asia; *Duration* - 20 days,
- **Objective**: *OB1. Name* - evaluationg the effects of the gamified approach; *Details* - qualify the use of the suggested framework (SGM), within the context of "cross-cultural" development,
- **Criteria**: *CR1. Name* - performance; *Description* - the performance of a team of software engineers in systems development,
- **Questions**: *QU1. Description* - is the code quality higher in a gamified design group compared to a control group? *Hypothesis* - yes. *QU2. Description* - is the self-evaluation of team members in a gamified experiment group greater than in a control group? *Hypothesis* - yes,
- **Measures**: **For QU1 question** *ME1. Name* - LOC - Total lines of code; *Unit* - dimensionless dimension; *Indicators* - fewer lines of code. *ME2. Name* - Total defects; *Unit* - dimensionless dimension; *Indicators* - less bugs. *ME3. Name* - Density of duplication; *Unit* - percentage; *Formula* - (duplicate lines / total lines) x 100; *Indicators* - lower duplication density; *Instrument* - SonarQube (objective data). *ME4. Name* - Density of Comments; *Unit* - percentage; *Formula* - (commented lines / total lines) x 100; *Indicators* - higher density of comments; *Instrument* - SonarQube (objective data). *ME5. Name* - Coverage of unit tests; *Unit* - percentage; *Formula* - (number of methods tested / number of methods) x 100; *Indicators* - higher coverage rate; *Instrument* - SonarQube (objective data). **For QU2 question** *ME6. Name* - Average self-evaluation on a Likert scale; *Unit* - dimensionless dimension; *Formula* - arithmetic average of grades from 1 to 10 among team members; *Indicators* - highest average score; *Instrument* - questionnaire (subjective data),
- **Rounds**: *RO1. Description* a team of 8 software engineers from different cultures (CCSDT) is formed, for the development of APIs in .NET/C#, applying the suggested framework (SGM); *Duration* - 10 days. *RO2. Description* - a team of 8 software engineers is formed in the same context, without the application of the suggested framework (SGM); *Duration* - 10 days,
- **Sample**: each sample is unique in the evaluation, and there are 16 participants in the study, half being an experimental sample, and the other being a control sample. The only demographic data collected was related to the places where these participants are natural, resided or reside, and graduated as software engineers,
- **Collection**: all collected data were summarized following the established metrics,
- **Analysis**: quantitative analyzes were performed on all collected data, taking into account the defined questions. Almost all hypotheses were confirmed in this isolated experiment, with the exception of one (test coverage - ME5),
- **Finding**: with the exception of coverage, most metrics seem to support the hypothesis that the SGM project is better. However, as the sample size of the teams is not large enough and the cultural mix is not exactly the same between the two projects, the authors have some reservations about drawing concrete conclusions.

The first inadequacy found in the framework was the absence of an attribute "formula" in the entity "Measure", which had not been included in the first study.

The relationship between collections, analyzes and population was the second and last inadequacy found. Before, each collection had its analysis, but it was an analysis for each instance of the population, which proved to be incorrect. It was observed that, in practice, the collections were grouped among all of the sample in one round, and analyzed by comparing the collections grouped between the two rounds.

Thus, a new attribute called "formula" was inserted in the generic entity "Measure", and the relationship between the analysis and the collection was updated, allowing the same analysis to study more than one collection, and that the same collection can be analyzed several times. The entity "Finding" was only abstracted after the second proof of concept. However, the PoC was reapplied in the first study to evaluate the use of this new entity.

2) Second Primary Study [19]: The second instantiation of the evaluation framework for this primary study was carried out as follows, considering the values defined for the attributes of each entity:

- **Gamification**: *Context* - teaching and evaluation of software engineering; *Description* - each participant had to assume a specific role (manager, trainer, developer, on-site customer or usability expert) on their team. Each week students were given a specific challenge. Each challenge was about one of the extreme programming practices and how they were applying it to the project,
- **Evaluation**: *Research Method* - case study; *Sample Description* - 50 students of software engineering; *Duration* - 14 weeks,
- **Objectives**: *OB1. Name* - evaluating the gamified strategy; *Details* - the students' perception of the gamified strategy and its application in the class. *OB2. Name* - evaluating the effects of the gamified approach; *Details* - the effects of challenges on programming practices,
- **Criteria**: **For OB1 objective** *CR1. Name* - Satisfaction; *Description* - approval of the course style, learning experience and software project experience. *CR2. Name* - Engagement; *Description* - engagement in the software project. **For OB2 objective** *CR3. Name* - Performance; *Description* - optimized programming, quality of documentation, extreme programming practice. *CR4. Name* - Engagement; *Description* - engagement in the software project,
- **Measures**: *ME1. Name* - Student's perception of learning; *Unit* - percentage; *Formula* - (number of responses per unit of the Likert scale / number of responses) x 100; *Indicators* - the 5-3 scale units, which represent the best perceptions, have a higher average than the other scale units; *Instrument* - questionnaire (subjective). *ME2. Name* - Student's perception of engagement; *Unit* - percentage; *Formula* - (number of responses per unit of the Likert scale / number of responses) x 100; *Indicators* - the 5-3 scale units, which represent the best perceptions, have a higher average than the other scale units; *Instrument* - questionnaire (subjective). *ME3. Name* - Note in practice of XP; *Unit* - dimensionless dimension; *Indicators* - the adoption of extreme

programming practices during challenges, and after challenges; *Instrument* - teacher evaluation. *ME4. Name* - Number of commits; *Unit* - dimensionless dimension. *ME5. Name* - LOC - total lines of code; *Unit* - dimensionless dimension,

- **Rounds**: *RO1. Description* - monitoring the weekly performance of a course, and each challenge required the application of an aspect of extreme programming, using a gamified approach; *Duration* - 6 weeks. *RO2. Description* - monitoring the weekly performance of a course, without the gamified approach; *Duration* - 8 weeks,
- **Sample**: all teams participated in the evaluation, in both rounds. There were 5 teams of 10 volunteer students,
- **Collection**: only the data referring to the student's perception metrics were grouped and presented (ME1 and ME2). However, the study does not clarify at what point in the evaluation these grouped data were collected,
- **Analysis**: *AN1. Type of analysis* - quantitative; *Description* - 79.35% of students rated their learning success as very good or good with gamification compared to other similar university courses in programming without gamification. Participants got used to this new form of subject, they started to evaluate their learning and coding performance even better than before. *AN2. Type of analysis* - quantitative; *Description* - students were more engaged with the topic covered by the weekly challenge,
- **Finding**: programming was a growing practice throughout the course, but refactoring and testing declined after a few weeks. When a topic was revised, the practices related to it resumed the growth of its adoption.

An inadequacy was found in the framework when analyzing this case study. The absence of a "Finding" entity that abstracts the grouping of analyzes and the conduct of the discussion on the result of an evaluation. With the addition of "Finding", the two instances of the framework were revised and corrected.

However, some information was not given or is not clear in this primary study. First, there was no related objective to the engagement criterion used. The author does not specify whether the engagement criteria was related to the engagement of participants in the gamification dynamics, or if it is the engagement in the course activities. Second, no questions were defined, i.e., the author defines the evaluation criteria and determines which measures will be used, without justifying the need for these measures. Third, it is not possible to identify whether the summary data was collected from the experiment population or from the control population. And finally, only the data related to the users' perception measures were summarized in this study, but the author makes reference to data related to other measures (Note in extreme programming practice, commits and LOC), in the analysis of these data, and in their findings.

D. Discussion

Our main contribution in this study was the design of a evaluation framework for gamification in software engineering teaching and training, based on the findings on the state-of-the-art and the state-of-practice of gamification reported by Monteiro et al. [1]. One implication of this is providing a set of procedures and generic entities that, if specified, allow the comparison of different results obtained in different studies on gamification.

We noticed that all entities considered inadequate at some point in the PoC were revised and adjusted to the two primary studies evaluated. However, this is a preliminary evaluation performed in only 2 studies for PoC. Therefore new opportunities for improvement can be found with additional PoC studies. However, these inadequacies tend to decrease very quickly, since the design of this framework uses the same information provided by these studies, summarized in [1] previous systematic mapping.

The finding of this PoC studies is the impact of the lack of some information related to the evaluation in the second case study [19]. The non-specification of evaluation questions makes the choice of evaluation measures less objective. The failure to present sufficient detail on data, which are used as an argumentative basis for the auhtors' analysis, reinforces the need for standardization and transparency of the evaluation procedures. Thus, although we do not define which abstract entities need to be instantiated in order for the evaluation to be understood, we observe that the lack of some information may compromise the verification of internal validity.

VI. Threats To Validity

This section discusses the possible threats to the validity of this study and the actions taken to solve the validity problems. We use the structure proposed in [20].

A. Construct Validity

In order to minimize the risks that the framework would not be able to adequately contain the instances of each primary study, the PoC was carried out and its results were discussed among the three authors, two of whom are specialist researchers in the field. The evaluation was consolidated if there was unanimity on the results discussed.

B. Internal Validity

During the analysis process, the studies were classified based on our judgment. However, despite the double verification, some studies may have been classified incorrectly, mainly with regard to the relationship between the evaluation criteria and the objectives of the evaluation.

C. External Validity

It is possible that the study of the design of the evaluation framework for gamification does not include all relevant studies on software engineering and evaluation gamification for this context. To mitigate this risk, we use the data summarized in [1], which in turn expands two other systematic mappings [21] [2], to abstract the entities that are present among the findings reported there. We also use the GQIM model to structure the entities found in the mapping, which in turn is used for the same purpose in a validated framework design study [9].

D. Conclusion Validity

To ensure the validity of the conclusion of our study, throughout Subsection C of Section V we present the full instances of the framework for the two primary studies used in the PoC exposing the results generated directly from the data and discussing the explicit observations and trends in the Subsection D of Section V.

VII. Conclusion and Future Work

This study describes an evaluation framework for gamification in software engineering education and training. We have identified four evaluation steps and 14 abstract entities that classify the information in an evaluation. We analyzed the findings found in [1] and the GQIM model to design the framework, and we evaluate its suitability using PoC. We consulted two primary studies [18] [19], checked if the entities in our framework could be used to categorize information on gamification evaluation in these two studies, and made the necessary adjustments in the framework. Thus, we provide a set of relevant procedures and information to the generation of significant empirical data for the study of the impact of gamification on software engineering.

For future work, we plan to analyze the impacts that each entity of the framework has on the evaluation of gamification, and to evaluate the items of this framework through peer review and case studies.

ACKNOWLEDGMENT

This work was supported by CAPES (Coordination for the Improvement of Higher Education Personnel) from the granting of a master's scholarship number 88887.485345/2020-00.

REFERENCES

[1] R. H. Barbosa Monteiro, M. R. de Almeida Souza, S. R. Bezerra Oliveira, C. dos Santos Portela, and C. E. de Cristo Lobato, "The diversity of gamification evaluation in the software engineering education and industry: Trends, comparisons and gaps," in *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, 2021, pp. 154–164.

[2] M. Souza, L. Veado, R. T. Moreira, E. Figueiredo, and H. Costa, "A systematic mapping study on game-related methods for software engineering education," *Information and Software Technology*, vol. 95, pp. 201–218, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584917303518

[3] S. Jiang, "A review of the effectiveness of gamification in education," *SSRN Electronic Journal*, 2016. [Online]. Available: https://doi.org/10.2139/ssrn.3163896

[4] S. Bai, K. F. Hew, and B. Huang, "Does gamification improve student learning outcome? evidence from a meta-analysis and synthesis of qualitative data in educational contexts," *Educational Research Review*, vol. 30, p. 100322, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1747938X19302908

[5] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: Defining "gamification"," in *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, ser. MindTrek '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 9–15. [Online]. Available: https://doi.org/10.1145/2181037.2181040

[6] T. Dal Sasso, A. Mocci, M. Lanza, and E. Mastrodicasa, "How to gamify software engineering," in *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, Feb 2017, pp. 261–271.

[7] G. P. Gasca-Hurtado, M. C. Gómez-Álvarez, M. Muñoz, and J. Mejía, "Proposal of an assessment framework for gamified environments: a case study," *IET Software*, vol. 13, no. 2, pp. 122–128, Apr. 2019. [Online]. Available: https://doi.org/10.1049/iet-sen.2018.5084

[8] W. Ren, S. Barrett, and S. Das, "Toward gamification to software engineering and contribution of software engineer," in *Proceedings of the 2020 4th International Conference on Management Engineering, Software Engineering and Service Sciences*, ser. ICMSS 2020. New York, NY, USA: Association for Computing Machinery, 2020, p. 1–5. [Online]. Available: https://doi.org/10.1145/3380625.3380628

[9] G. Petri, C. G. V. Wangenheim, and A. F. Borgatto, "Meega+: Um modelo para a avaliação de jogos educacionais para o ensino de computação," *Revista Brasileira de Informática na Educação*, vol. 27, no. 03, pp. 52–81, Dec. 2019. [Online]. Available: https://doi.org/10.5753/rbie.2019.27.03.52

[10] V. Basili, G. Caldiera, and H. D. Rombach, "The goal question metric approach," 1994.

[11] R. E. Park, W. B. Goethert, and W. A. Florac, *Goal-Driven Software Measurement - A Guidebook*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA 15213, 1996. [Online]. Available: https://resources.sei.cmu.edu/asset_files/Handbook/1996_002_001_16436.1996_002_001_16436.pdf

[12] M. Zhu and A. I. Wang, "Model-driven game development: A literature review," *ACM Comput. Surv.*, vol. 52, no. 6, Nov. 2019. [Online]. Available: https://doi.org/10.1145/3365000

[13] S. K. Sripada, Y. R. Reddy, and S. Khandelwal, "Architecting an extensible framework for gamifying software engineering concepts," in *Proceedings of the 9th India Software Engineering Conference*, ser. ISEC '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 119–130. [Online]. Available: https://doi.org/10.1145/2856636.2856649

[14] R. M. Parizi, "On the gamification of human-centric traceability tasks in software testing and coding," in *2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA)*, June 2016, pp. 193–200.

[15] A. Neto, M. Borges, and L. Roque, "Characterizing proof-of-concept practices using the lens of context engineering," in *Information Systems Development: Information Systems Beyond 2020 (ISD2019 Proceedings)*, Toulon, France: ISEN Yncréa Méditerranée, Aug. 2019. [Online]. Available: https://aisel.aisnet.org/isd2014/proceedings2019/ISDMethodologies/4

[16] C. Jobin, S. Hooge, and P. Le Masson, "What does the proof-of-concept (POC) really prove? A historical perspective and a cross-domain analytical study," in *XXIXème conférence de l'Association Internationale de Management Stratégique (AIMS)*, Online, France, Jun. 2020. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02570321

[17] J. Guimarães, "Método para manutenção de sistema de software utilizando técnicas arquiteturais." Ph.D. dissertation, Escola Politécnica, Universidade de São Paulo, 2008. [Online]. Available: https://doi.org/10.11606/d.3.2008.tde-29012009-134316

[18] I. Chow and L. Huang, "A software gamification model for cross-cultural software development teams," in *Proceedings of the 2017 International Conference on Management Engineering, Software Engineering and Service Sciences*, ser. ICMSS '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 1–8. [Online]. Available: https://doi.org/10.1145/3034950.3034955

[19] B. S. Akpolat and W. Slany, "Enhancing software engineering student team engagement in a high-intensity extreme programming course using gamification," in *2014 IEEE 27th Conference on Software Engineering Education and Training (CSEE T)*, April 2014, pp. 149–153.

[20] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*. Springer, jun 2012. [Online]. Available: https://www.xarg.org/ref/a/3642290434/

[21] O. Pedreira, F. García, N. Brisaboa, and M. Piattini, "Gamification in software engineering – a systematic mapping," *Information and Software Technology*, vol. 57, pp. 157–168, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584914001980