Quizzes (as a tool for self-regulated learning) in Software Engineering Education

José Bernardo Rocha
Instituto Superior Técnico, Universidade de Lisboa
INESC-ID
Lisboa, Portugal
jose.b.g.rocha@tecnico.ulisboa.pt

Luis Felipe Coimbra Costa
Programa de Engenharia de Sistemas e Computação
Coppe/UFRJ
Rio de Janeiro, Brazil
luisfcosta@cos.ufrj.br

Rui Prada
Instituto Superior Técnico, Universidade de Lisboa
INESC-ID
Lisboa, Portugal
rui.prada@tecnico.ulisboa.pt

António Rito Silva
Instituto Superior Técnico, Universidade de Lisboa
INESC-ID
Lisboa, Portugal
Rito.Silva@tecnico.ulisboa.pt

Daniel Gonçalves
Instituto Superior Técnico, Universidade de Lisboa
INESC-ID
Lisboa, Portugal
daniel.goncalves@inesc-id.pt

Pedro Correia
Instituto Superior Técnico, Universidade de Lisboa
INESC-ID
Lisboa, Portugal
pedro.miguel.correia@tecnico.ulisboa.pt

***Abstract*—This paper discusses how quizzes are applied within the field of software engineering learning and how quizzes can help self-regulate student learning. For this, a systematic mapping that selected the most relevant studies on the use of quizzes in education was performed, aiming to clarify their relationships and mutual impacts. Our analysis shows that student engagement and quiz work is a prominent learning solution for increasing motivation in and out of the classroom. We found that quizzes can be applied in software engineering and through generic quizzes, online Quizzes, pop-Quizzes, gamified Quizzes, in quiz games or, alternatively, as an exercise in creating quizzes. However, we did not find approaches to the use of quizzes, explicitly containing Zimmerman's cyclic model, only some of the model's activities in isolation and not explicitly. We argue that sharing the quizzes will raise the potential for then to be use as a self-regulation tool in software engineering education. We describe the steps taken by a Software Engineering Gamification project to create a effective tool for creating and sharing software engineering quizzes. Our next requirement to be implemented in the project will be the application of self-regulation of learning containing the three phases of the Zimermman cyclic model.**

***Index Terms*—Quizzes, Self-regulated learning, Software Engineering Education**

I. INTRODUCTION

Student's way of learning changes, particularly with the use of technological tools [1]–[3]. Despite this, the types of active learning are primarily intended to make the learner at the center of the learning process and not necessarily focused on the use of technology. Examples of this occurs even in the approaches used since the time of Aristotle (384 BC-322 BC) [4], such as a class where students are seated in a wheel, discussing and reflecting, among all, a specific subject.

One of the technological elements used as a resource for learning are quizzes. Using quizzes for learning has been around for many years since it was first discussed [5]. Quizzes have been used in high education, as an assessment tool and to support other educational methods or techniques in various different fields, including engineering courses [6]–[8].

Grimstad [9] found results indicating that students could use quizzes to improve their exam performance. For this performance improvement to occur, Bangert-Drowns [10] analyzed that feedback is related to learning and concluded that students learn more if they receive the correct answer only after answering a question. It also found that corrective feedback is better than simply telling students that they were right or wrong, meaning that it is important to guide the student to the content areas where they need review, or further study. This is due to the fact that, according to the author, it promotes a strategy with aspects about preparation, execution, feedback, and self reflection.

The importance of this strategy, which relates to the three phases of the self-regulation of learning, and how it can be applied to the educational area of software engineering, drives our intention to explore existing works that use quizzes as a learning tool. Additionally, one of our main motivations for executing a systematic mapping is that it allows us to clarify architectures, models, and frameworks related to quiz usage in software engineering education. In particular, we intend to assess whether a multi-institutional shared repository of software engineering quiz questions would be a desirable contribution to software engineering education.

For this, our study presents the following research questions:

- **RQ_1 - Are Quizzes used in Software Engineering Education?**
- **RQ_2 - How are Quizzes used in Software Engineering Education?**
- **RQ_3 - Are Quizzes used in Software Engineering Education for the Self-regulation of Learning?**

- **RQ$_4$ - How are Quizzes used in Software Engineering Education for the Self Regulation of Learning?**
- **RQ$_5$ - How can we improve the use of Quizzes in Software Engineering Education?**

In order to answer these question we performed a mapping review into five sections, beginning with the introduction (section 1) and then progressing as follows: Section 2 presents a brief review of quizzes; Section 3 includes the concepts and methodology used in this work; section 4 covers the results and our research analysis; And finally, in section 5, our conclusions are presented.

II. Education and Quizzes

Anderson [11] discussed and analyzed questions about educational processes. These processes included sources of motivation, like self-efficacy beliefs, delayed gratification, attributions, values, and interests. As well as sources of meta cognition like goal setting, strategy use, self-monitoring, and self-evaluation.

Recently, due to the latest developments in technological resources, methods, dynamics, and motivational elements, education has been transforming traditional teaching methods into new learning concepts, such as the inverted classroom and self-regulated learning.

The inverted classroom is a technology-supported pedagogy. Bishop and Verleger [12] define this type of learning as a direct, computer-based individual instruction outside the classroom. In addition, according to the authors, this process is also carried out through video lectures and interactive group learning activities within the classroom.

As a result of all these transformations in education, the current view of the student's task in education has shifted to discovering how to learn new knowledge, rather than creating unique truths and knowledge [3]. By reflecting on their actions and how to find new knowledge, the learner can discover for himself how to overcome his own challenges.

Berge [4] compares the practice of the inverted classroom to the dialogue-based approach practiced in ancient Greece where people obtained learning through real life challenges and activities, sharing their own thoughts and opinions in order to find solutions to their problems.

Zimmerman and Risemberg [13] explained the implications of the different components of self-regulated learning, stressing that the proposed tasks should allow students to make personal and thoughtful decisions with the intention of regulating their learning processes. Based on these implications, the cyclical model of Zimmerman [14] was established (a schematic representation is included in Fig. 1). According to Zimmerman, self-regulated learning aims to define a student's learning process and motivational beliefs based on three self regulation phases: Forethought, Performance and Self-Reflection [15].

Self-regulated learning was also considered important in social forms of learning, such as seeking outside help from people other than the teacher. The main question is whether a student demonstrates personal initiative and adaptive ability.

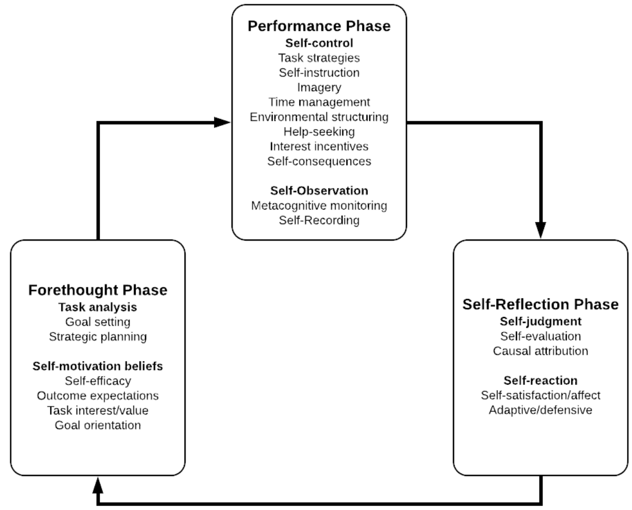

Fig. 1. Zimmerman Cyclic Model.

These qualities of learners derive from advantageous motivational feelings and beliefs [14]

According to Anderson et al. [16], there are two types of assessment. The first type is summative assessment where information is collected post-learning in order to determine how well the student has learned the material. The other type is formative assessment, which is concerned with collecting information about learning, and designed to improve the quality or quantity of learning.

Quizzes are typically used to take a student's knowledge and skills assessment tests. The quizzes should not be understood only as summative assessments, but understood as a learning experience for the students, evaluating the effectiveness of their study strategies and test preparation [1]. As such, they are one of the elements that can be applied in education to improve self-regulated learning.

It is important to consider that learning how to use the study material is as important as learning the material itself. This way students will evaluate their strategies and worry about their performance, improving them accordingly [2]. To this end, Zimmerman suggests that students should first be asked to express their confidence in the ability to solve a problem before they start trying to solve it and re-evaluate their confidence after solving it. After that, they should be asked to write an error analysis of all problems that they did not solve correctly. By doing the first activity, students quickly learn to study a problem before trying to solve it; while doing the second, allows them to acquire learning solution strategies [17].

Ottenhoff [18] states that the literature on self-regulated learning, metacognition and assessment proposes various activities and tasks that students can perform before, during and after exams. Some activities and assignments make students more aware of what they have learned and not learned, while other activities involve students in evaluating test preparation strategies and developing more effective strategies. They all help improve students' focus on their learning.

According to Nilson [1], quizzes can support self-regulated learning by providing opportunities for several Activities and Assignments that help with self-regulated learning. These include Activities and Assignments to Prepare for Exams, Activities During an Exam and Activities and Assignments After Exams and Quizzes (a schematic representation is included in Fig. 2).

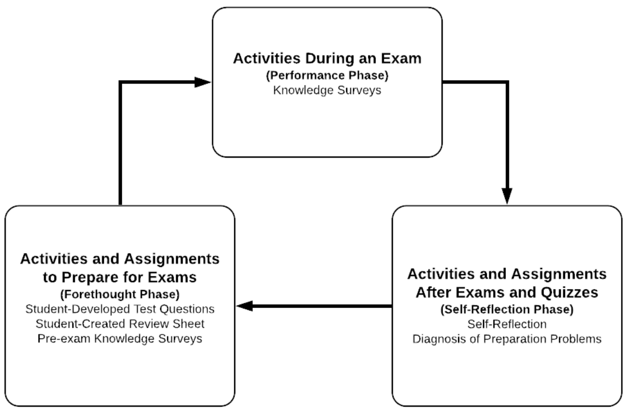

Fig. 2. Self Regulated Learning Activities Supported by Exams / Quizzes.

Before Exams, students can prepare themselves by studying and improving on their weaker areas of knowledge, but less time consuming options exist. By creating Student-Developed Test Questions (multiple-choice or other types of questions), students can review the material and assess the importance of different parts of the material. The creation of a Student-Created Review Sheet that maps major content areas allows the student to estimate the percentage of the exam that will be made up of that area. This review sheet can be further refined by listing what students believe they should be able to do or demonstrate and use this review sheet to prepare themselves. Students can elaborate Pre-exam Knowledge Surveys (questionnaires that ask them to rate their confidence in their ability to answer questions or perform tasks related to the course). This familiarizes students with the quiz/exam's objective and reveals content and skill areas in which they are weak or strong, allowing them to narrow the focus of their study and encouraging goal setting and self-testing.

During an Exam, students can be asked to rate their confidence in their ability to solve a problem before and after tackling it. Incorporating a knowledge survey into an exam can allow the results to be weighed by the student's confidence in them.

After an Exam, there are several opportunities for self-regulated learning. Students can be asked to reflect on how prepared they were for a quiz and on the effectiveness of their preparation method, allowing them to review, and reconsider their preparation methods and current ability. The return of a graded quiz provides the opportunity for students to resolve problems they previously missed or solve similar problems. If this is coupled with pre-quiz or pre-exam self-assessment of their confidence levels and of the effectiveness of their study method it allows for impressive leaps in student learning. Students can be asked questions at the end of an exam to estimate their performance, to describe their preparation, and to evaluate the difficulty in different parts of the exam. This makes students reflect and predict their performance, evaluate their preparation and the effectiveness of their preparation methods, reassess their confidence after receiving the graded exam, and see if their preparation strategies are working. A formalization of the reassessment of a student's methods and confidence is possible by having the students answer a series of questions in writing after they see their graded exam, thus allowing students to develop a study game plan for the next exam based on the results of the previous exam. By using post-test analysis, students are able to diagnose the patterns behind their errors, and, thus, obtain the type of help they need.

Although there are many theoretical and practical questions about how to maximize the benefits of quiz use, disrupting learning with quiz questions is beneficial because they can improve student engagement [19]. Several educational institutions adopt clickers, which allow instructors to administer multiple choice questionnaires at any time during a lecture, with immediate feedback to teachers and students [20]. According to Harley, there is a learning benefit for students, provided by the action of testing their knowledge, from getting feedback, and the interactive nature of the clicker procedure. For teachers, there is the benefit to assessment due to having the ability to maintain continuous assessment of students' understanding and predict their learning retention after reading the material [19].

With the understanding that the quizzes can be applied to education, despite the apparent challenge and overhead required to prepare and administer quizzes in an educational context, this paper aims to investigate how quizzes can be beneficial for self-regulated learning, in the context of software engineering classes.

III. METHODOLOGY

The research process used in this work is based on the systematic review process research in software engineering [21], [22] and was conducted by using the Parsif.al [1] platform.

Our process began with a systematic mapping that allows us to evaluate and select the most relevant studies on the use of quizzes to help self-regulation of learning. Our intention is to identify the use of quizzes in self-regulation of learning in the context of undergraduate students, in software engineering learning.

We explore how the teaching of software engineering uses quizzes, what are their types, if it is used as a tool for self-regulation of learning, and which positive and negative results were found. For us it is important to understand how they could be used in different ways. All literature review

[1] https://parsif.al/

procedures - such as description of what was planned, strategy selection, research conduct, and final report - are described below.

A. Research Planning

Our research plan was prepared containing: objectives, questions to answer, search, and selection, strategies, quality assessment and results extraction.

Our research planning began by defining the protocol for the systematic mapping. We began by defining the objectives of our research:

- Analyze scientific publications that address Software Engineering Teaching;
- Analyze scientific publications that study the use of Quizzes in Teaching (and particularly as a self-regulating pedagogical tool);
- Develop a Review and / or Mapping of the literature on "Software Engineering Teaching and Quizzes";
- Justify / Motivate the creation / implementation of a Software Engineering Shared Quiz repository

Our next step was defining the Population, Intervention, Comparison, Outcome, Context (PICOC):

- Population - "quizzes"
- Intervention - self-regulation of learning
- Comparison - initiative OR model OR framework OR activities OR methods OR techniques
- Outcome - support OR improve OR optimize OR aid OR help OR specification
- Context - Software Engineering Education

This was followed by defining the research questions according to the following rationale:

- **RQ_1 - Are Quizzes used in Software Engineering Education?** - The purpose of this question is to verify whether quizzes are used as a learning tool in software engineering education.
- **RQ_2 - How are Quizzes used in Software Engineering Education?** - The purpose of this question is to understand what types of quizzes were applied and how they were used in Software Engineering Education.
- **RQ_3 - Are Quizzes used in Software Engineering Education for the Self-regulation of Learning?** - The purpose of this question is to understand which areas of research, related to self-regulation of learning, were affected by the use of quizzes.
- **RQ_4 - How are Quizzes used in Software Engineering Education for the Self Regulation of Learning?** - This question seeks to identify which stages of self-regulation of learning, based on Zimmerman's cyclical model, were applied, their results and their implementation failures.
- **RQ_5 - How can we improve the use of Quizzes in Software Engineering Education?** - This question seeks to identify possible improvements to the current usage of quizzes in software engineering education.

Regarding Keywords and Synonyms, we selected the following keywords but chose not to define any synonyms in order to avoid getting any results that fell outside the scope of this work:

- Quiz
- Self Regulation Learning
- Software Engineering Education

We chose the Scopus database as our source. In order to search for articles, we used articles from the last 5 years in our selected sources. We defined the following search string *"quiz" AND "Software Engineering Education"*. We defined several inclusion and exclusion criteria so that we could accept or reject the studies that we had obtained from our search. These are shown in **Table I**.

TABLE I
INCLUSION AND EXCLUSION CRITERIA FOR STUDIES

Inclusion Criteria	**Exclusion Criteria**
The study presents possible methods and techniques	The study approaches quizzes outside the context of software engineering
The study presents possible implications (advantages and disadvantages)	The study is a chapter of a book or an editorial
The study presents questions related to our research	The study is not available for reading
-	The study is not related to the research questions
-	The study was not published in English

After finishing defining the protocol, we defined a Quality Assessment Checklist that allowed us to evaluate the quality of each of the studies that had been accepted. The checklist was composed of two questions that we found adequate for the objectives of our research:

- Does the article address self-regulation learning through the use of quizzes?
- Does the article address the types of quizzes used in software engineering education?

We defined three answers for the questions and their respective weight. These are shown in **Table II**.

TABLE II
QUALITY ASSESSMENT CHECKLIST QUESTION ANSWERS AND RESPECTIVE WEIGHT

Answer	**Weight**
Yes	2.0
Partial	1.0
No	0.0

The maximum score was automatically computed (4.0) and we chose a Cutoff Score (which allows us to discard any study that was bellow the chosen score) of 3.0.

So that we could systematize the data extraction of the studies, we created a Data Extraction Form that could help us finding the answers to our research questions with the following fields:

- **Type of Application of Quizzes** - This field's objective is to extract data on how quizzes were applied to Software

Engineering (types of quizzes). It allows us to answer $\mathbf{RQ}_1$ and $\mathbf{RQ}_2$.

- **Context where Quizzes were applied** - This field's objective is to extract data on the context where Quizzes were applied (what type of course or system were quizzes applied to, and when were they applied). It will help us find answers to $\mathbf{RQ}_1$ and $\mathbf{RQ}_2$.
- **Phases of the Zimmerman Cyclical Model implicitly or explicitly considered** - This field's objective is to extract data on what phases of the Zimmerman Cyclical Model were involved or looked at in the study due to the application of quizzes. It will assist us in finding answers to $\mathbf{RQ}_3$ and $\mathbf{RQ}_4$.
- **Types of Activities (Before, During or After Exam/Quiz) implicitly or explicitly considered** - This field's objective is to extract data on what type of Activities were involved or looked at in the study due to the application of quizzes. Like the previous field, it will help us find answers to $\mathbf{RQ}_3$ and $\mathbf{RQ}_4$.
- **Positive Results from the Application of Quizzes** - This field's objective is to extract data on what were the positive results of the application of Quizzes found by the study. We will be able to look at the positive aspects of the application of quizzes in software engineering that can help us reason on $\mathbf{RQ}_5$.
- **Negative Results from the Application of Quizzes** - This field's objective is to extract data on what were the negative results of the application of Quizzes found by the study. By being aware of the negative aspects, we can attempt to minimize them in order to answer $\mathbf{RQ}_5$.

B. Conducting the Research

Study selection used a two-step filtering procedure: The first step involved reading the studies and filtering them according to whether an individual study is related to our research; while the second step involved a scoring system, in which the pre-selected studies from the previous step were analyzed in detail to determine the degree to which they responded to the survey (i.e.: fully, partially or not all). Then, the extraction was performed by consolidating the relevant information contained in the studies. Finally, an analysis of the result of the previous step was performed so that we could answer the research questions.

The research we conducted had the following in-order steps:

1) *Search* - In this step, we search the selected repositories for the previously defined search string. This allows us to obtain a list of studies that matched our search string, which we then use in the next step.
2) *Import Studies* - This step allows us to import the studies' meta-information (abstract, authors, date, etc.) in order to maintain a database to manage the next steps.
3) *Study Selection* - During this step, we accept or reject the studies in our database according to the criteria we defined during the planning.
4) *Quality Assessment* - Here, we apply the previously defined Quality Assessment Checklist to our database's studies that were accepted in the Study Selection step in order to score them individually.
5) *Data Extraction* - We use the previously created Data Extraction Form to extract the data relevant to our research from the studies in our database that scored above 3.0 in the previous step.
6) *Data Analysis* - We analyse the data that resulted of applying our research methodology.

IV. RESULTS

In this section, the results of the review are presented and analyzed.

A. Search

The execution of our Search query return a total of 68 studies, of which we had no duplicates.

B. Study Selection

Our analysis of the studies obtained during the Search made us reject 46 studies because they were not relevant to our research questions.

C. Quality Assessment

We then scored the remaining 22 studies using the previously defined Quality Assessment Checklist which resulted in us keeping the 13 studies. An overview of the results can be shown in **Table III**.

TABLE III
OVERVIEW OF THE QUALITY ASSESSMENT SCORES OF THE STUDIES

Score	Number of Studies
0	1
1	2
2	6
3	1
4	12

We now present a brief description of the studies that were kept after finishing the Quality Assessment:

- The Federal University of Minas Gerais, in Brazil, presents and evaluates an open course in Introduction to Software Engineering with students who have studied the discipline for two consecutive years (2012 and 2013) [8]. Although there is no clear description or study on the process of self-regulation of learning, we can understand that the process of performing quizzes during all modules of the course, together with viewing videos, provides greater attention on the part of students regarding the individual effort to carry out the tasks. Altogether 16 quizzes were used throughout the course, and the teachers understood that the quizzes allow instructors to gain better control of students who need help in specific modules and can also help them to obtain better results.
- The University of the West Indies, in Jamaica, introduced the use of a competitive system, through a programming platform, called Hacker Rank TM as a mechanism to automatically assign grades to an introductory course

on design and analysis of algorithms [23]. The course is assessed through a combination of writing and / or programming tasks (30%), two online tests (10%), one intermediate exam (10%) and final exam (written) (50%). The method used for competitive evaluations among students was based on scores related to the level of difficulty of the questions. For the authors, more frequent tests increase performance. On the other hand, they indicate that a more careful analysis of the negative effects of gamification on students with low self-confidence is necessary. We did not find explicit references on self-regulation of learning.

- A working group, composed of researchers from various universities, produced a report that gave rise to a very detailed systematic review, containing hundreds of articles and more than 50 pages [24]. The objective was to obtain an overview of the literature on the introductory part of programming. This study addresses the student, teaching, curriculum and assessment. The authors explored trends, highlighted advances in knowledge in this area in the last 15 years and indicated possible directions for future research. Among its conclusions, it is noteworthy that a large proportion of publications provide few details about the context in which the report (activity / study / experience) is carried out. In the authors' view, for a reader to determine whether the results of the work are transferable to their own teaching context, it is important to know details about the student population and the teaching context. It was noted that in may of the studies examined, the measurement, the operationalization and the effects described were not reported. Finally, the authors state that the studies did not provide sufficient detail to allow study replication, or to obtain valid and reliable results which would allow better informed recommendations for the practice of teaching. Despite these limitations, in the context our study, we found three study references that included an approach to the use of quizzes. The first of these states that quizzes used to determine whether students completed preparation before class meetings are reported that no have benefit. The second states that gamification to motivate students to complete online quizzes and daily feedback for programming exercises through assessment tools, were effective approaches for engaging students. The third study, which includes the use of quizzes, indicates that the use of think-pair-share (TPS) with a traditional lecture format in a class showed that students who learned from TPS believed that it helped them learn. In a subsequent questionnaire, the TPS group performed significantly better than the control group. We only found a single reference to the term self-regulation of learning, where the student code was combined with protocols that relied on thinking out loud in order to investigate the self-regulation of learning in programming.
- Another systematic review reports important information about intelligent tutoring systems designed to teach computer programming [25]. Regarding our study, we found no evidence related to the self-regulation of learning and we found a reference to the use of multiple choice, true and false, or short answer quizzes, though no results were presented. and the use of quizzes together with exercises interactive presents a systematic review that reports important information about intelligent tutoring systems designed to teach computer programming.
- Ogawa, conducted a pilot study, using the inverted classroom methodology, which increased student engagement and task through video classes and quizzes related to the content of the videos [26]. Students were gathered in the classroom once a week for a 75-minute lecture and subsequently watched a video of approximately 25 minutes online, developed by the teacher and at the end, the student should answer the quizz of 10 questions based on the content of the video. In addition, students also have the option of providing feedback on quiz questions and adding their own video to better illustrate the concepts.
- With regard to software engineering education, using a game model that assesses the benefits of using digital and non-digital games, we found an article that evaluates these games through the MEEGA model [27]. This analysis was based on data collected from 43 case studies using MEEGA, evaluating 20 games for teaching software engineering, involving a total of 723 students. The authors' analysis indicates that digital and non-digital games can have a positive effect in learning software engineering, providing a pleasant and engaging environment for students and providing motivation to learn. No use of quizzes found.
- In the area of software verification and validation, a teaching area of software security, we found an article where the authors propose a partnership with the industry to develop 14 hours of course modules, containing class exercises, case studies and didactic videos on the content [28]. The authors make a proposal for the student's understanding to be assessed through quizzes, tests, assignments and a learning survey
- We found a study describing a course that provides an introduction to computational thinking and object-oriented concepts before introducing programming, through a MOOC called, LOOP: Learning Object-Oriented Programming [29]. Besides using normal tests, interactive exercises were developed and used to allow students to experiment and interact directly with the concepts being taught. Additionally, programming exercises that allowed constructive feedback were also implemented by using an integrated development environment together with an automatic classification system. According to the authors, the overall feedback was positive.
- An action research to that attempts to improve students' design skills was described by one of the studies. The proposal was designed in order to teach software engineering design to large groups [7]. The authors' approach combined projects with weekly tests, tests and active learning

tasks. The Quizzes used questions marked with underlying concepts and cognitive levels, thus enabling teachers to identify common misunderstandings. The authors found that tests defined at levels of analysis and synthesis seemed to promote better software with respect to design skills. The active learning tools created helped to correct common misunderstandings, by providing immediate and holistic information such as comments. The new teaching approach helped to improve student retention, satisfaction and performance substantially.

- An approach that used the inverted Classroom model in teaching software engineering by using structured discussions, weekly quizzes and workshops with invited experts from the technology sector [6]. The authors used surveys and interviews with students and teachers to review this teaching method. The evidence found was that there was a measurable increase in the amount of discussion. However, it was not possible to prove an increase in the quality of the discussions.
- EduJudge is a project that aims to integrate the "UVA Online Judge," an online instructor of programming with an important number of problems in an educational environment which consisted of the Moodle e-learning platform and a competitive learning tool [30]. The ultimate goal was to provide new learning strategies that motivated students and presented programming as an easy and attractive challenge. EduJudge has been tried and tested in programming courses in three different degrees of engineering. The levels of motivation and satisfaction of the students were analyzed, as were the effects of the EduJudge system on the academic results of the students. The results indicated that students and teachers have found that, among other multiple benefits, the EduJudge system facilitated the learning process. In addition, the experience also showed an improvement in students' academic results.
- A problem when teaching in higher education classrooms is the lack of support for the interaction between students. Some authors propose a lecture game concept that can improve communication and motivate students through more interesting lectures [31]. It is a multiplayer quiz game, called Lecture Quiz. When comparing the evaluation data of the second version with the first version of the Lecture Quiz, it was possible to identify that both surveys showed that the concept of the Lecture Quiz is a suitable to improve lectures in most aspects, and that the Lecture Quiz had been improved in many ways since the beginning of its implementation. According to the authors, the results were interesting in terms of exploring this area of research.
- The same author who carried out an analysis on the benefits of digital and non-digital games used in software engineering education [27], points out in this article [32] an analysis of the same MEEGA evaluation model for teaching Software Project Management. This analysis was based on data collected from 27 case studies, evaluating 11 different games, involving a population of 562 students. The results provide evidence that digital and non-digital games contribute positively to the perception of learning by students, as well as promoting social interaction, and being considered relevant to student learning and promoting fun.

D. Data Extraction

In our review, we obtained relevant information about quizzes in software engineering courses [8], as well as systematic reviews of works that applied quizzes in this educational context, such as [24], [25]. We also found studies that evaluate outcomes in an inverted classroom setting that work with self-regulating learning, such as [26] and [6]. Student learning practices, such as [7] and [30]. One of these studies, [31] presented a proposal for a lecture game concept that can improve communication and motivate students through more interesting lectures and while not clearly describing how students can improve their self-regulation of lecture. Learning, this study can provide important questions concerning students' motivation to learn, and interest in the game, in this case a multiplayer quiz game called Lecture Quiz.

Table IV shows the data extracted related to the **Type of Application of Quizzes**. We identified the following types: Quiz Games, games where students answer quizzes in order to get high scores or compete with each other; Online Quizzes, quizzes that are administered online; Pop-Quizzes, quizzes that the students need to answer without having any previous warning; Gamified quizzes, quizzes that students are incentivized to anwer by using gamification design elements; Quiz Creation, where students need to create quizzes that can, potentially, be used by other students or themselves as review material; Generic Quizzes, the standard type of quizz that is taken in class through traditional means.

TABLE IV
TYPES OF APPLICATION OF QUIZZES

Types of Application of Quizzes	**Number of Studies**	**Studies**
Quiz Games	4	[27], [30]–[32]
Online Quizzes	4	[8], [23], [24], [29]
Pop-Quizzes	1	[24]
Gamified Quizzes	1	[24]
Quiz Creation	1	[26]
Generic Quizzes	5	[6], [7], [25], [26], [28]

We found a variety of **Contexts where Quizzes were applied**. We have separated these contexts into the following categories: **In-Lecture vs Outside-Lecture**, **Type of Course** (Traditional, Online, Hybrid, or Intelligent Tutor System) and **Course Subject**. **Table V** shows the **In-Lecture vs Outside-Lecture** contexts that were found for application of quizzes.

Table VI shows the **Type of Course** contexts that were found for the application of quizzes.

Table VII shows the **Course Subject** contexts that were found for the application of quizzes.

TABLE V
CONTEXT WHERE QUIZZES WERE APPLIED: IN-LECTURE VS OUTSIDE-LECTURE

In-Lecture vs Outside-Lecture	**Number of Studies**	**Studies**
In-Lecture Quizzes	6	[7], [25], [27]–[29], [31]
Outside-Lecture Quizzes	7	[6]–[8], [23], [24], [26], [30]

TABLE VI
CONTEXT WHERE QUIZZES WERE APPLIED: TYPE OF COURSE

Type of Course	**Number of Studies**	**Studies**
Traditional Courses	6	[7], [23], [24], [27], [28], [31]
Online Courses	4	[8], [26], [29], [30]
Online Courses Hybrid Course	1	[8]
Intelligent Tutor System	2	[7], [25]
Inverted Class Course	3	[6], [23], [26]

TABLE VII
CONTEXT WHERE QUIZZES WERE APPLIED: COURSE SUBJECT

Course Subject	**Number of Studies**	**Studies**
Algorithms	2	[23], [26]
Programming	5	[24]–[26], [29], [30]
Software Design	1	[7], [31]
Software Engineering	3	[6], [8]
Software Project Management	2	[27], [32]
Software Security	1	[28]

TABLE VIII
PHASES OF THE ZIMMERMAN CYCLICAL MODEL IMPLICITLY OR EXPLICITLY CONSIDERED

Phase	**Number of Studies**	**Studies**
Forethought	6	[6], [23], [24], [29]–[31]
Performance	6	[23], [24], [26], [27], [29], [30]
Self-Reflection	6	[7], [8], [23], [24], [27], [29]

Regarding **Phases of the Zimmerman Cyclical Model (implicitly or explicitly) considered**, the articles did not analyse quizzes directly through the perspective of the Zimmerman Cyclical Model, but they approached them nonetheless. **Table VIII** shows the results.

Some of the studies attempted to leverage quizzes for certain **Types of Activities**. The results are shown in **Table IX**.

The studies showed the following **Positive Results from the Application of Quizzes**: Educational Games contribute positively to the Learning Experience [27], Quizzes allow better self-assessment [8], Pop-Quizzes improve students performance, and Gamified Quizzes improve student's engagement [24], Quizzes and the creation of quizzes increased Task Performance [26], Quiz Games can promote an engaged experience to the players [32]. Users liked the interactivity provided by online quizzes, [29]. Quizzes that allow the diagnostic of common errors allow students to improve, and the classification of quiz questions into concepts and cognitive levels allows the course's staff and students to identify problem areas [7], Quizzes were effective as a motivator for watching lectures in an inverted classroom [6], Quiz Games improved the academic outcomes of students [30], and In-Lecture Quiz Games made students provide closer attention and had a positive effect on learning [31]. We found motivational aspects in learning that were applied in some works as a way to improve learning, including active learning, reported by [6], [8], [27], [28], [32]. As a way of generating student feedback, it was cited by [8], [29], [30], student involvement, including motivation, was found in [6], [23], [24], [29]–[31]. In addition, work has been found to improve student attitude [24], improve student experience in [24], [32], improve student satisfaction in [30], and improve student communication in [31].

The **Negative Results from Application of Quizzes** highlighted by the studies were the following: Preparation Quizzes before a class had no benefit [24], and Weekly graded quizzes can increase student anxiety [6].

E. Data analysis

After synthesizing the data extracted, we answered our research questions:

RQ$_1$ - Are Quizzes used in Software Engineering Education? - We can answer this question clearly with a yes. Quizzes are used in software engineering education.

RQ$_2$ - How are Quizzes used in Software Engineering Education? - Quizzes can be applied in a variety of forms. We found that they can be applied as Generic Quizzes, as Online Quizzes, as Pop-Quizzes, as Gamified Quizzes, in Quiz Games or, alternatively, as an exercise in the creation of Quizzes. The contexts where we saw them being applied were also varied, and we found 3 important dimensions for the context: were they applied In-Lecture or Outside-Lecture; what type of course/learning system were they applied on (Traditional, Online, Hybrid or Intelligent Tutor System); and what was the course subject where they were applied (Algorithms, Programming, Software Design, Software Engineering, Software Project Management or Software Security).

RQ$_3$ - Are Quizzes used in Software Engineering Education for the Self-regulation of Learning? - The data tells us that yes, quizzes are used in Software Engineering Education as self-regulated learning. In the studies, we found examples that approached or focused on each of the different phases of the Zimmerman Cyclical Model. Additionally, concerning self-regulated learning Activities supported by quizzes, we found all the types of self-regulated learning activities that quizzes support, even if they did so implicitly and not explicitly.

RQ$_4$ - How are Quizzes used in Software Engineering Education for the Self Regulation of Learning? - The extracted data shows that it can be used in order to support different phases of the Zimmerman Cyclical Model, and

TABLE IX
TYPE OF ACTIVITIES (BEFORE, DURING OR AFTER EXAM/QUIZ) IMPLICITLY OR EXPLICITLY CONSIDERED

Activity Type	**Number of Studies**	**Studies**
Activities Before an Exam	7	[6], [8], [23], [24], [26], [30], [31]
Activities During an Exam	1	[30]
Activities After an Exam	6	[7], [8], [23], [24], [27], [29]

different types of self-regulated learning Activities. However, as mentioned previously, this was done implicitly in most cases and not explicitly.

RQ_5 - How can we improve the use of Quizzes in Software Engineering Education? - We have found no clear answer to this question. The studies examined showed some negative aspects that need to be taken into account carefully when using Quizzes. We need to make sure that the application of Quizzes does not increase student anxiety, and we also need to carefully examine how to apply Preparation Quizzes so that it can produce the adequate benefits. It is also important to consider how to maximize the benefits to the student's performance, engagement, motivation, attitude, the benefits from having a feedback mechanism, and the benefits of using them as a tool for self-assessment and diagnostic. A tentative, but unsupported answer to this question, may simply be, facilitate the application quizzes to software engineering education. This can be achieved by having a shared repository of quizzes so that they can be easily deployed. This idea can be further expanded by supplying the tools required to manage and administer those quizzes. A limitation that we have found in the studies was that they did not approach the Zimmerman Cyclical Model and the previously mentioned Activities Supported by Quizzes explicitly. We suspect that a deployment of Quizzes in Software Engineering Education that approaches or focuses on these aspects explicitly might obtain stronger positive results and possibly minimize the negative ones.

V. QUIZZES TUTOR

The Quizzes Tutor[2] system is a repository for the management of multiple choice questions, which can be categorized into topics, and their aggregation into quizzes that has been used in Software Engineering courses for both, assessment and self-assessment. Therefore, besides already having a corpus of quizzes that came from running the course in previous years, an important aspect of our tool is that, not only does it support the teacher role (creating and editing quizzes and quiz questions) it also has a component that supports the student role (taking the quiz). note that the questions selected for the automatically generated quizzes for students are chosen considering the students previous interactions with the system, in terms of previous, correct and incorrect, answers. One of the more interesting aspects, was that, in order to better integrate the Zimmerman Cyclic Model and the Self Regulated Learning Activities Supported by Exams / Quizzes, students are allowed to take non-mandatory quizzes so that they can practice the course material.

The system code is being used as a learning tool, in the software engineering classes, which fosters a motivational context, where the students implement new functionalites in the system while learning the development of web applications. Currently, the system is being enriched with functionalities that (1) allow students to ask a clarification about a question they have answered, and the clarifications to become available after answering; (2) allow students to submit question proposals to be analyzed and approved by the teacher; and (3) allow students to organize question competitions on the course topics. With these functionalities we intend to foster self regulated learning where: (1) Self-Reflection Phase; (2) Performance Phase; (3) Forethought Phase

After this initial test run of our tool and repository, they will be shared across multiple institutions and have their content enhanced with additional Software Engineering course subjects, such as Test-Driven Development and Formal Specifications.

The repository code is publicly available in a GitHub repository[3].

VI. CONCLUSION

This paper examined the impact of quizzes on self-regulated learning in order to identify the best applications of quizzes in the field of software engineering, and to do this we mapped and analyzed recent work. The review used a structured approach with well-defined steps. Some studies have gone through filtering steps before being approved for further analysis. The results of the analysis showed the potential benefits of using quizzes as a tool for self-regulated learning.

Thus, we found in these studies common and proposed challenges to address the use of quizzes in the context of software engineering and to improve students self-regulated learning. It was also possible to identify failures, mainly related to a correct application of Zimmerman's cyclic model, which provides the basis and process to support self-regulated learning for students. This relationship between learning and motivation has a great impact on educational environments, and as a consequence, we can say that quizzes can be used as a way of assessing students, but also to stimulate the participation, motivation and learning of these students. We believe that current studies can be better explored and with more research in this field, especially concerning:

- Self-regulating models, methods and techniques of learning supported by the use of quizzes
- Research relating to the impact, in the form of learning, on software engineering
- Use of quizzes shared among various institutions

[2]https://quizzes-tutor.tecnico.ulisboa.pt

[3]https://github.com/socialsoftware/quizzes-tutor

- Procedures for assessing the quality and validity of quiz questions based on scientific criteria

We are argue that, due to the previously referred benefits and due the bootstrapping effort required in order to create, review and use quizzes, a standard body of peer-reviewed questions and answers that can be reused in different faculties that can better reflect the state of the art and current knowledge of the topic being taught. It can be a useful contribution to software engineering education, due to it conferring the ability to potentially analyze student skills at various universities, which can enable teachers to better understand the weaknesses of the curriculum and the problem areas that may cause problems for students, as well as providing a better understanding of what those problems are.

Acknowledgment

This work was developed for the IMPRESS project funded by EU Erasmus+ Programme, grant nr. 2017-1-NL01-KA203-035259 and by the Fundacção para a Ciência e a Tecnologia, under project UIDB/50021/2020.

References

[1] L. Nilson and B. J. Zimmerman, *Creating Self-Regulated Learners: Strategies to Strengthen Students' Self-Awareness and Learning Skills*. Stylus Publishing, 2013.

[2] L. D. Fink, *Creating Significant Learning Experiences: An Integrated Approach to Designing College Courses*, 2nd ed. San Francisco: Jossey-Bass, 2013.

[3] E. S. Ebert and R. C. Culyer, *School: An Introduction to Education*, 3rd ed. Belmont, CA: Cengage Learning, 2013.

[4] Z. L. Berge, *Computer Mediated Communication and the Online Classroom: Distance Learning*, M. P. Collins, Ed. Cresskill, NJ: Hampton press Cresskill, 1995.

[5] V. T. Mawhinney, D. E. Bostow, D. R. Laws, G. J. Blumenfeld, and B. L. Hopkins, "A comparison of students studying-behavior produced by daily, weekly, and three-week testing schedules," *Journal of Applied Behavior Analysis*, vol. 4, no. 4, pp. 257–264, 1971. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1310703/

[6] M. J. Herold, T. D. Lynch, R. Ramnath, and J. Ramanathan, "Student and instructor experiences in the inverted classroom," in *2012 Frontiers in Education Conference Proceedings*. IEEE, 2012, pp. 1–6.

[7] C. Thevathayan and M. Hamilton, "Imparting software engineering design skills," in *Proceedings of the Nineteenth Australasian Computing Education Conference*. ACM, 2017, pp. 95–102.

[8] E. Figueiredo, J. A. Pereira, L. Garcia, and L. Lourdes, "On the evaluation of an open software engineering course," in *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*. IEEE, 2014, pp. 1–8.

[9] K. Grimstad and M. Grabe, "Are online study questions beneficial?" *Teaching of Psychology*, vol. 31, no. 2, pp. 143–146, 2004.

[10] R. L. Bangert-Drowns, C.-L. C. Kulik, J. A. Kulik, and M. Morgan, "The instructional effect of feedback in test-like events," *Review of educational research*, vol. 61, no. 2, pp. 213–238, 1991.

[11] R. C. ANDERSON, "Some Reflections on the Acquisition of Knowledge," *Educational Researcher*, vol. 13, no. 9, pp. 5–10, Nov. 1984. [Online]. Available: https://doi.org/10.3102/0013189X013009005

[12] J. L. Bishop and M. A. Verleger, "The flipped classroom: A survey of the research," in *ASEE national conference proceedings, Atlanta, GA*, vol. 30, 2013, pp. 1–18.

[13] B. J. Zimmerman and R. Risemberg, "Chapter 4 - Self-Regulatory Dimensions of Academic Learning and Motivation," in *Handbook of Academic Learning*, ser. Educational Psychology, G. D. Phye, Ed. San Diego: Academic Press, 1997, pp. 105–125. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780125542555500053

[14] B. J. Zimmerman, "Chapter 2 - Attaining Self-Regulation: A Social Cognitive Perspective," in *Handbook of Self-Regulation*, M. Boekaerts, P. R. Pintrich, and M. Zeidner, Eds. San Diego: Academic Press, 2000, pp. 13–39. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780121098902500317

[15] ——, "Investigating Self-Regulation and Motivation: Historical Background, Methodological Developments, and Future Prospects," *American Educational Research Journal*, vol. 45, no. 1, pp. 166–183, 2008. [Online]. Available: https://doi.org/10.3102/0002831207312909

[16] L. W. Anderson, D. R. Krathwohl, P. W. Airasian, K. A. Cruikshank, R. E. Mayer, P. R. Pintrich, J. Raths, and M. C. Wittrock, *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Abridged Edition*, 1st ed. New York: Pearson, 2000.

[17] B. J. Zimmerman, A. Moylan, J. Hudesman, N. White, and B. Flugman, "Enhancing self-reflection and mathematics achievement of at-risk urban technical college students," *Psychological Test and Assessment Modeling*, vol. 53, no. 1, pp. 141–160, 2011.

[18] J. Ottenhoff, "Learning How to Learn: Metacognition in Liberal Education," *Liberal Education*, vol. 97, pp. 28–33, 2011.

[19] A. F. Healy, M. Jones, L. A. Lalchandani, and L. A. Tack, "Timing of quizzes during learning: Effects on motivation and retention." *Journal of Experimental Psychology: Applied*, vol. 23, no. 2, p. 128, 2017. [Online]. Available: https://psycnet-apa-org.ez29.capes.proxy.ufrj.br/fulltext/2017-10883-001.pdf

[20] M. K. Smith, W. B. Wood, W. K. Adams, C. Wieman, J. K. Knight, N. Guild, and T. T. Su, "Why Peer Discussion Improves Student Performance on In-Class Concept Questions," *Science*, vol. 323, no. 5910, pp. 122–124, 2009. [Online]. Available: https://science.sciencemag.org/content/323/5910/122

[21] B. Kitchenham and P. Brereton, "A systematic review of systematic review process research in software engineering," *Information and Software Technology*, vol. 55, no. 12, pp. 2049–2075, 2013. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0950584913001560

[22] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering – A systematic literature review," *Information and Software Technology*, vol. 51, no. 1, pp. 7–15, 2009. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0950584908001390

[23] D. Coore and D. Fokum, "Facilitating Course Assessment with a Competitive Programming Platform," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. ACM, 2019, pp. 449–455.

[24] A. Luxton-Reilly, I. Albluwi, B. A. Becker, M. Giannakos, A. N. Kumar, L. Ott, J. Paterson, M. J. Scott, J. Sheard, and C. Szabo, "Introductory programming: a systematic literature review," in *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*. ACM, 2018, pp. 55–106.

[25] T. Crow, A. Luxton-Reilly, and B. Wuensche, "Intelligent tutoring systems for programming education: a systematic review," in *Proceedings of the 20th Australasian Computing Education Conference*. ACM, 2018, pp. 53–62.

[26] M.-B. Ogawa, "Evaluation of flip-flop learning methodology," in *International Conference on Learning and Collaboration Technologies*. Springer, 2018, pp. 350–360.

[27] G. Petri, C. G. von Wangenheim, and A. F. Borgatto, "Quality of games for teaching software engineering: an analysis of empirical evidences of digital and non-digital games," in *Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track*. IEEE Press, 2017, pp. 150–159.

[28] S. Acharya, "Infusing software security in software engineering," in *ASEE Annual Conference and Exposition, Conference Proceedings*, 2017.

[29] J. Krugel and P. Hubwieser, "Computational thinking as springboard for learning object-oriented programming in an interactive MOOC," in *2017 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2017, pp. 1709–1712.

[30] E. Verdú, L. M. Regueras, M. J. Verdú, J. P. Leal, J. P. de Castro, and R. Queirós, "A distributed system for learning programming on-line," *Computers & Education*, vol. 58, no. 1, pp. 1–10, 2012.

[31] B. Wu, A. I. Wang, E. A. Børresen, and K. A. Tidemann, "Improvement of a Lecture Game Concept-Implementing Lecture Quiz 2.0." in *CSEDU (2)*, 2011, pp. 26–35.

[32] G. Petri, A. Calderón, C. G. von Wangenheim, A. F. Borgatto, and M. Ruiz, "Games for Teaching Software Project Management: An Analysis of the Benefits of Digital and Non-Digital Games." *J. UCS*, vol. 24, no. 10, pp. 1424–1451, 2018.