Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 3, 2026

GAMIFICATION IN SOFTWARE TESTING EDUCATION: A SYSTEMATIC LITERATURE REVIEW

Khadija Ali Asghar*[1], Kalsoom Fatima[2], Usama Ahmed[3]

*[1,2]*Department of Software Engineering, Government College University Faisalabad*
[3]*Department of Software Engineering*

[*1]khadijaaliasghar80@gmail.com, [2]kalsoomfatima982@gmail.com, [3]usamaahmed@gcuf.edu.pk

DOI: http://doi.org/10.5281/zenodo.21126855

Keywords
Gamificationn, Software Testing Education, Systematic Literature Review, Based Learning, Mutation Testing, Unit Testing, Student Engagement, Learning Outcomes, Serious Game

Article History
Received: 25 January 2026
Accepted: 10 March 2026
Published: 24 March 2026

Copyright @Author
Corresponding Author: *
Khadija Ali Asghar

Abstract

Software testing is an essential but often underestimated aspect of computer science, with computer science graduates often lacking in testing skills, which is why software testing is considered a challenging yet poorly valued subject area in computer science education. The systematic application of game design elements in environments outside of games (gamification) has proven to be a very promising pedagogical approach to solve these motivational deficiencies and, however, there is no consolidated evidence base that systematically characterizes the application of gamification in software testing education and the measurable impact on students' motivation and learning outcomes. This systematic literature review (SLR) is designed to find, analyze, and synthesize the previous empirical and design-based studies on gamification in software testing education and give a structured overview of the tools and the dominant game mechanics, the methodologies employed in the studies, and the evidence of the effectiveness of gamification in software testing education. Using a predefined Boolean search string in accordance with the PRISMA 2020 guidelines [33] a structured search was conducted in IEEE Xplore and Google Scholar databases for the following keywords: gamification, game-based learning, and serious games, as well as software testing, test-driven development, mutation testing and unit testing. 74 records were retrieved, with 11 of them being duplicates, and 63 unique studies were available and were included and excluded based on predefined inclusion and exclusion criteria, after which 44 studies were identified for final synthesis which took place between 2015 and 2025. The results indicate that the gamification has been applied in various software testing sub-domains, such as mutation testing (n = 9), unit testing (n = 7), and exploratory testing (n = 4), and many popular tools used to implement gamification exist, such as Code Defenders, GAMFLEW, CoverBot, Code Critters, TSGame, and IntelliGame. The most commonly used game elements are competition mechanics, points and scoring systems, badges, leaderboards and serious game frameworks. The evidence-based research on the effectiveness of gamification is based on three types of studies controlled experiments, quasi-experimental studies, and longitudinal case studies and all of these studies show positive results in terms of student engagement, motivation, and time-on-task, with some studies reporting statistically significant improvements in

test quality metrics. There is, however, limited evidence on long-term retention and transfer of skills. This review shows that gamification is a potential and emerging pedagogical method for the field of Software Testing education and identifies the need for further rigorous experimental designs, multi-institutional replicas and longitudinal studies.

1. INTRODUCTION

1.1 Background and Motivation

Software is pervasive and forms the foundation of the critical systems in today's healthcare, financial, transportation, infrastructure and public-service industries [1]. It is generally accepted that software testing is a key way to ensure software reliability before its deployment [2], [3], [4] . The repercussions of poor testing can be catastrophic: In one year, 3.7 billion people suffered from 606 major software failures, which resulted in $1.7 trillion in financial losses, according to Tricentis report[5]. Garousi and Zhi [6] have reported that in their experience, testing is often an afterthought in academic programs, and Kasurinen et al. [7] have reported that testing is not well educated in the curriculum of many practising engineers. Minhas and Barrett [8] established that there were few dedicated software testing courses and testing was often included in a short module within other software engineering courses.

Students' experience of testing is consistently seen as a dull, repetitive and uneventful activity from a cognitive point of view, in comparison to programming and system design activities [9], [10]. This negative perception leads to a lack of motivation to participate in the testing activities, and graduates with low levels of competence in testing when they enter the workforce [8], [10]. There is a consistent focus on covering more ground in constructive activities programming, design, and architecture, than in verification/validation in curricula [4], [9]. The solution to this problem lies in pedagogical interventions that can help to change the students' attitude to testing, boost their motivation and enhance the quality of the skills acquired as a result of academic training in the field of testing.

1.2 Gamification as a pedagogical strategy

The use of gamification as a pedagogical strategy is introduced. Gamification as a pedagogical strategy is introduced. According to Deterding [11] gamification can be formally defined as the use of game design elements in non-game setting. This definition is different from those of other researchers who consider gamification a complete educational game experience [12] or game-based learning using existing games as learning tools [12]. The foundation of gamification is the Self-Determination Theory (SDT) [13] which states that there are three basic psychological needs: autonomy, competence, and relatedness, and their fulfillment increases intrinsic motivation. Kapp [14] claimed that the game mechanics such as progressive challenge, immediate feedback, achievement recognition and competition are structurally aligned with these needs. Twenty-four empirical gamification studies were reviewed by Hamari et al. [12] and most of the studies showed positive impacts on motivation, engagement and enjoyment.

In computer education, Pedreira et al.[15] have performed the first systematic mapping of gamification in software engineering on 49 studies. Fraser [10] expressed the initial vision of turning software testing into a game, which is characterized as a competitive interaction between different parties that are trying to find as many bugs and errors as possible in the other party's code. This was most notably realised in CodeDefenders [16], [17] which was deployed at various educational institutions in Europe, and later in other tools, such as GAMFLEW [18], CoverBot [19], Code Critters, TSGame and IntelliGame [20].

1.3 Research Gap, Objectives, and Contributions

Although there is increasing primary research, there is no comprehensive systematic synthesis in the field. So far, secondary studies on software engineering education gamification have considered this topic in general without concentrating on software testing [15], [21], [22] and the latest empirical studies [23], [24], [25] are after the previous reviews. This study fills the gap for an SLR that focuses on gamification in software testing education, as there is no PRISMA-compliant SLR on this topic.The lack of PRISMA-compliant SLRs focused only on the topic of gamification in software testing education is filled by this study: [26], [27].

The SLR aims to achieve four goals: (1) to compile all peer-reviewed literature published in the last 12 years (2014-2025); (2) to describe the gamification tools and mechanics; (3) to summarise empirical research on engagement and learning outcomes; and (4) to outline the research gaps and future research questions. Original contributions: First separate SLR on this topic; Structured taxonomy of 16 tools; Calibrated evidence synthesis (44 studies); 7-point research agenda.

1.4 Paper Structure

The related literature is discussed in Section 2. The methodology is presented in Section 3. The results are organised by seven research questions in Section 4. The findings and the theoretical implications are explored in Section 5, as are threats to validity. The conclusion and future research directions are given in Section 6. References are at the end of the paper.

2. RELATED WORK

This section summarizes the existing literature with four thematic areas that are supported with visualisations from the 44 studies which have been included and compared. Strong grasp of the theoretical nuances behind the gamification, a solid understanding of the conceptual aspect of gamification.

2.1 The Conceptual Foundations of Gamification

The canonical definition of gamification by Deterding et al. [11] clearly separates the concept and serious games and game-based learning [14]. The motivation principles it is based upon are Self-Determination Theory [13] and the game mechanics are related to the psychological needs for competence, autonomy, and relatedness. Hamari et al. [12] conducted a review of 24 empirical studies, and found that most studies reported positive motivational effects, but noted that contextual moderators play an important role in determining the magnitude of the effects [12]. Kapp [14] compiled the evidence that the use of game-based learning improves knowledge retention, the transfer of skills, and motivates learners. Malone [28] contended that challenge, fantasy and curiosity are intrinsic properties of well-designed games which are essential factors for intrinsically motivating instruction.

2.2 Gamification in Software Engineering Education

Pedreira et al.[15] performed the first systematic mapping of gamification in Software Engineering (SE) to find engagement and motivation outcomes as the ones most frequently reported in 49 studies. Souza, et al. [29] proposed a mapping of game-based techniques to SWEBOK knowledge areas which further validates the software testing's gamification because of the students' reported reluctance. In a systematic review, which included 68 studies in SE education and training, Di Nardo et al. [30] identified points, badges, and leaderboards as the most common mechanics, and noted that there was limited evidence that they persisted in the long run. Herculani et al. [31] conducted a tertiary study that verified the field's expansion and revealed that there were ongoing methodological issues such as small samples, single-institution designs and limited replication. Before this review, Fulcini et al. [20] conducted an in-depth review of the literature of gamified software testing tools, both in educational and industrial settings, which resulted in an extensive tool catalogue.

2.3 Gamification in Software Testing Education

Fraser [10] has provided the base vision of gamification of software testing, stating that the nature of the testing task fits well with the competitive nature of a game. The most widely investigated tool is CodeDefenders [32], [16], [17], [33], [25] which has been applied in several European universities and has been used with regularity and resulted in a steady increase of improvements. GAMFLEW [18] gamifies the concept of white-box testing, CoverBot [19] focuses on test coverage, Code Critters [34], [35] focuses on teaching mutation testing to younger learners, TSGame [36] and Testing Robot Challenge [37] address issues of test smells and unit testing, and the IntelliGame [38] uses 174 participants in a replicability study to gamify IDE-integrated testing.GAMFLEW [18] gamifies the concept of white-box testing, CoverBot [19] focuses on test coverage, Code Critters [34], [35] focuses on teaching mutation testing to younger learners, TSGame [36] and Testing Robot Challenge [37] address issues of test smells and unit testing, and IntelliGame [38] gamifies IDE-integrated testing with 174 participants in a replicability study. EMVille [39] addresses the equivalent mutant problem. However, the empirical evidence to support the positive effects of gamification in software testing education is controlled. According to Blanco et al. [23] and Martins de Jesus et al. [24] and Ren [40]controlled empirical evidence supports the positive effects of gamification in software testing education.

2.4 Gap Analysis

No previous secondary study has performed a detailed study focused on gamification in software testing education with a particular interest in applying the PRISMA guidelines. Fulcini et al. [44] did not review the gamified testing tools with a structured RQ without a structured RQ. All previous reviews are predated by the latest good quality evidence [23], [40], [17], [38], [35], [36]. This study closes this gap by compiling for the first time 44 primary studies [27], [26] in a systematic manner, following the PRISMA methodology.

The following comparative visualisations (figures 1-4 below) were created based on the data of all the 44 studies included in the study and visualise the publication trend, the frequency of gamification elements, the testing sub-domain distribution and research methodology distribution.

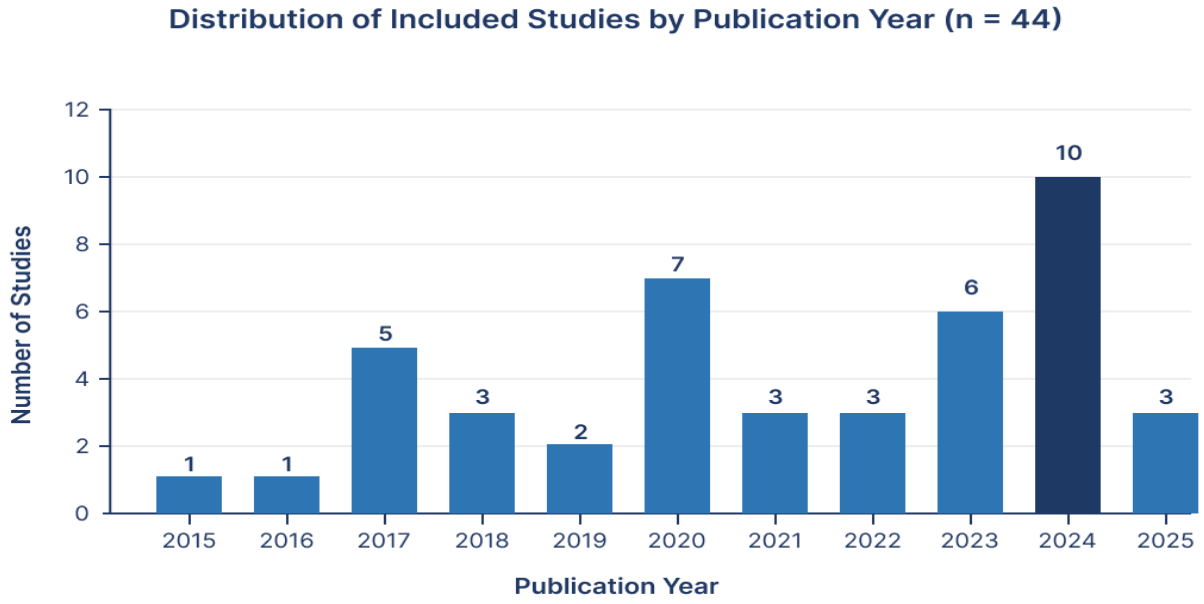


Figure 1: Distribution of included studies by publication

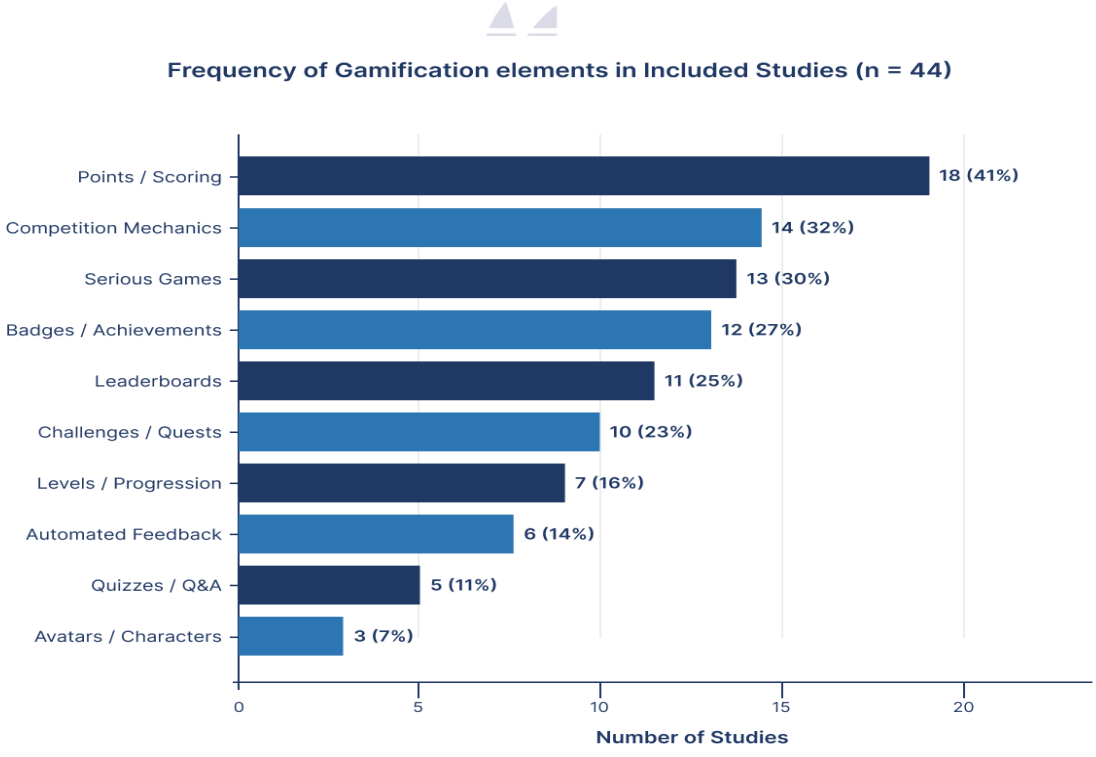


Figure 2: Frequency of gamification elements across included studies

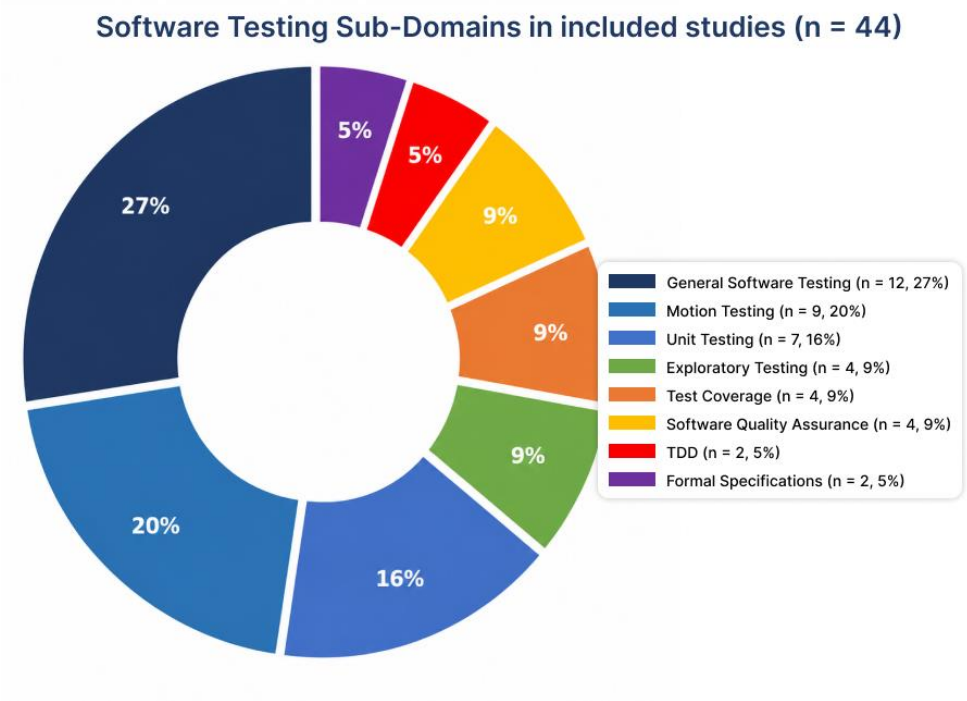


Figure 3: Software testing Sub-Doman in Included Studies (n=44)

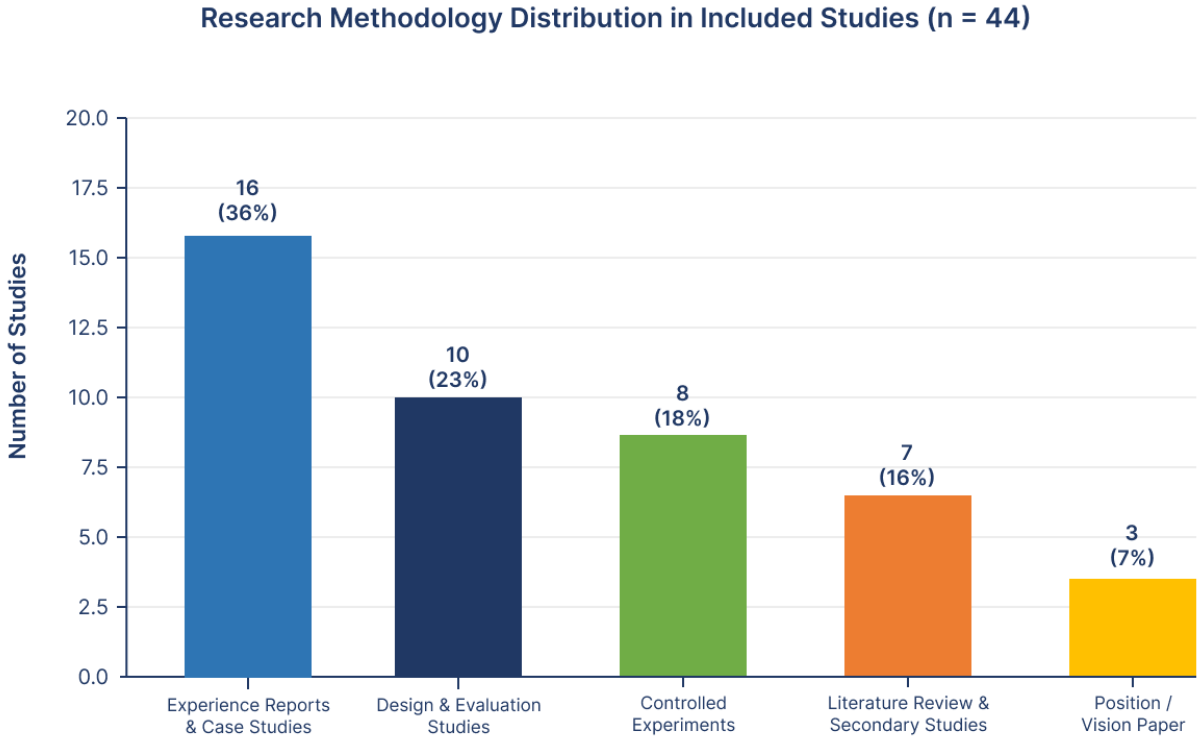


Figure 4: Research Methodology Distribution in Included Studies (n=44)

3. METHODOLOGY

This SLR is based on the guidelines of PRISMA 2020 [26] and Kitchenham and Charters [27] systematic reviews in software engineering. Five sequential stages were performed: protocol definition, search strategy, eligibility criteria, study selection and data extraction and synthesis.

3.1 Research Questions

A set of seven research questions (RQs) are used to guide the review and are shown in Table 1.

Table 1: Research Question Guiding the SLR

RQ No.	Research Question
RQ1	What gamification elements and mechanics are most commonly applied in software testing education?
RQ2	What software testing sub domain have been addressed through gamification?
RQ3	What tools and platforms have been developed to support gamification in software testing education?
RQ4	What is the impact of gamification on student engagement and motivation?
RQ5	What is the impact of gamification on student Learning outcomes and testing performance?
RQ6	What research methodologies and study designs have been employed?
RQ7	What limitations, challenges and open research issues are reported?

3.2 Search Strategy

The database that was chosen to be the primary was IEEE Xplore and the secondary database was Google Scholar. A Boolean search phrase was used, and searched across both DBs: OR "gamification" OR "game-based learning" OR "serious games" OR "game elements" AND "software testing" OR "test-driven development" OR "TDD" OR "Mutation testing" OR "Unit testing" AND "education" OR "learning" OR "teaching" OR "students" OR "training" Only publications in English and in the peer-reviewed format were searched. To streamline the research process, those studies published before 2014 were dropped during the eligibility screening process, because the current development of gamification in SE education began in 2014. Search was performed in May 2025 and prior to deduplication, 74 records were found in both the databases.

3.3 Eligibility Criteria

The inclusion criteria (IC) and exclusion criteria (EC) were applied and there were a total of six (Table 2).

Table 2: Inclusion and Exclusion Criteria

ID	Type	Criterion
IC1	Inclusion	Study applies gamification or game elements (points, badges, leaderboards, serious games, challenges) in an educational or training context.
IC2	Inclusion	Study focuses on software testing education or training as a primary topic (unit testing, mutation testing, TDD, test coverage, exploratory testing, SQA).
IC3	Inclusion	Published in a peer-reviewed journal, conference proceedings, or workshop.
IC4	Inclusion	Published in or after 2014.
IC5	Inclusion	Full text available in English.
IC6	Inclusion	Reports at minimum a design, implementation, or preliminary evaluation of a gamification approach.
EC1	Exclusion	Gamification applied only to general programming or SE without explicit software testing component.
EC2	Exclusion	Software testing appears solely as system validation, not as educational content.

EC3	Exclusion	Duplicate publication most complete version retained.
EC4	Exclusion	Secondary studies without specific gamification and testing education focus.
EC5	Exclusion	Technical or automated testing papers without educational gamification context.
EC6	Exclusion	Off-topic SE papers where testing is only incidentally mentioned.
IC1	Inclusion	Study applies gamification or game elements (points, badges, leaderboards, serious games, challenges) in an educational or training context.
IC2	Inclusion	Study focuses on software testing education or training as a primary topic (unit testing, mutation testing, TDD, test coverage, exploratory testing, SQA).

3.4 Study Selection Process

The selection process went through three stages in a sequential manner in line with PRISMA 2020 [26]. A total of 74 papers were identified in Stage 1 and 11 of these were duplicates, which were removed leaving 63 unique papers. No records were eliminated during the title and abstract screening in Stage 2, where the Boolean query was carefully defined with all 63 unique records screened. Stage 3 involved assessing full text of all 63 papers to eligibility criteria. Nineteen papers were excluded: EC1 (n=7), EC2 (n=2), EC3 (n=1), and EC4–EC6 (n=9). This resulted in 44 final studies that were used for synthesis.

3.5 Data Extraction and Synthesis

The student will use data and information to synthesize and draw inferences. The following information was captured in a structured extraction form: bibliographic information, gamification elements, testing sub-domain, research methodology, participant sample, key findings, reported outcome, limitations and quality indicators. The heterogeneity of the studies in terms of study design, intervention and outcome measures led to the use of a narrative synthesis approach. The quality assessment was carried out in accordance with a five criterion checklist adapted from Kitchenham et al. [27]; clarity of research objectives, methodological appropriateness, participant description, data collection rigour and transparency of limitations. The evidence was prioritised through quality scores, which provided a context for the evidence's strength.

4. RESULTS

The results are presented in the form of answers to the seven research questions. Each of the 44 studies included are directly cited in the relevant sub-sections.

This section contains quizzes on elements and mechanics of gamification.

4.1 RQ1 Gamification Elements and Mechanics

RQ1: Which gamification elements/techniques are frequently used in software testing education?

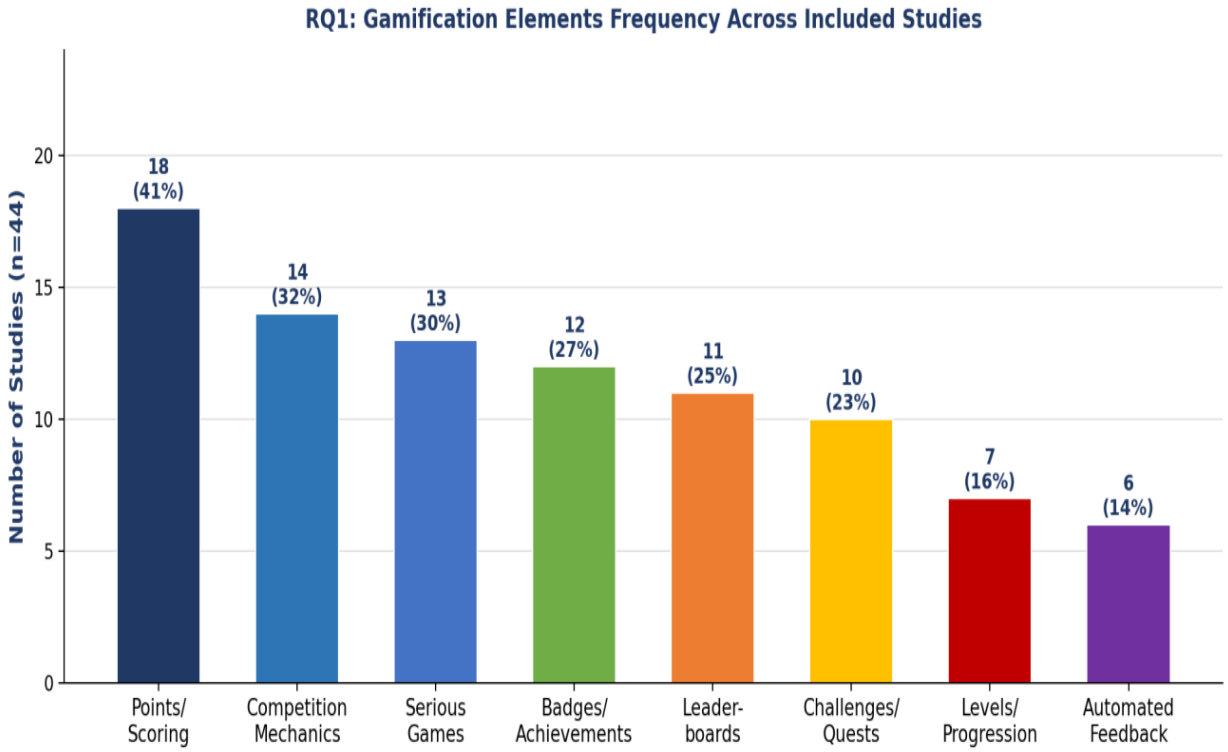


Figure 5: Frequency of gamification elements across 44 Included studies

The most studied themes followed are the competitor mechanisms (14 studies, 32%), serious game frameworks (13 studies, 30%), badges and achievements (12 studies, 27%) and leaderboards (11 studies, 25%). The most visible competition is in the CodeDefenders programme [10], [32], [33], [17], [41], [17] which uses an attack/defend model where students are attackers and introduce mutants and defenders and write tests to detect mutants. The GAMFLEW project [18] is based on serious game frameworks, the same are used in the project Code Critters [34] as well as in the project TSGame [36], Testing Robot Challenge [37] and EMVille [39] and the VR testing environment [42].

The project SOFTTY [43], [44] is based on serious game frameworks, the same are used in the project FormalZ [45]. There are several evidence-based approaches to gamification in the context of points and scoring systems, which are used widely in gamified course platforms [23], [24], [46] and are integrated into IDE [38] and are used in CI-gamification approaches [25]. While leaderboards are widely employed in CodeDefenders deployments [33], [17], [25] and gamified platforms [23], [24], [46] some research suggest that where students are ranked could actually demotivate lower-performing students [23], [40], [47]. The physical game formats card and board games are used by Soska et al. [43], [44] or by Lőrincz et al. [47] which provide low-technology solutions.

Gamification Element	Studies (n=44)	Primary Testing Context	Representative Studies
Points / Scoring	18 (41%)	General testing, Unit testing, TDD	[38], [23], [24], [40], [48]

Competition / Attack-Defend	14 (32%)	Mutation testing, Unit testing	[10],[32],[16],[33],[17],[41],[25],[49]
Serious Game Framework	13 (30%)	Mutation, White-box, Exploratory	[18],[34],[35],[36],[37],[39],[42],[43],[44]
Badges / Achievements	12 (27%)	General testing, IDE-integrated	[17],[38],[23],[24],[48]
Leaderboards	11 (25%)	Mutation testing, General testing	[16],[25],[17],[23],[24],[40]
Challenges / Quests	10 (23%)	Test coverage, SQA, White-box	[18],[19],[36],[37],[23],[41],[41]
Levels / Progression	7 (16%)	General testing, TDD, Exploratory	[23],[24],[46],[40],[50],[51]
Automated Feedback	6 (14%)	CI-integrated testing, IDE testing	[25],[38],[52],[48],[51]

4.2 RQ2: Software Testing Sub-domains

RQ2: What are the software testing sub-domains that have been tackled by using gamification?

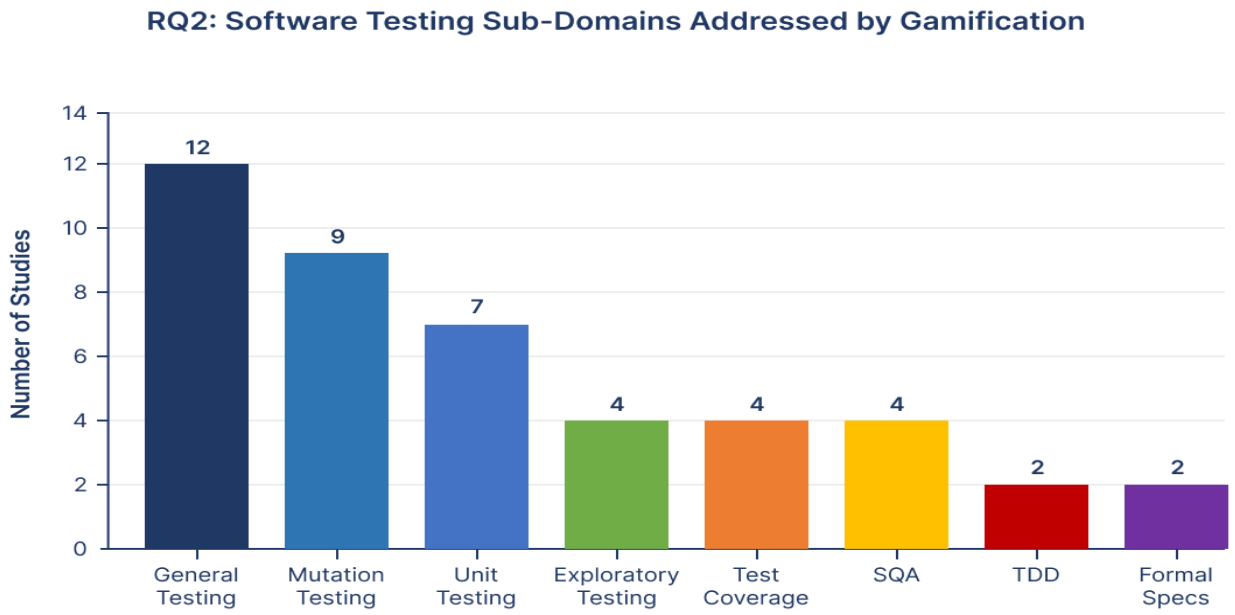


Figure 6: Software Testing sub-Domains Addressed by Gamification

General software testing accounts for the largest proportion (12 studies, 27%), including broad course interventions [86, 87, 88, 91, 92, 95, 104, 111, 112, 113] and secondary studies [105, 106, 109, 110]. As for the specific sub-domain, mutation testing is the most studied sub-domain with 9 studies (20%) dominated by CodeDefenders [71, 72, 73, 74, 75, 76, 77], Code Critters [80, 81] and EMVille [85]. In total, 7 studies (16%) are found that pertain to unit testing, these include UnitBrawl [102], Testable [90], Testing Robot Challenge [83], IntelliGame [84] and TDD gamification [89]. Exploratory testing appears in 4 studies [91, 92, 94, 96]; test coverage in 4 studies [78, 79, 101, 104]; software quality assurance in 4 studies [95, 103, 111, 114]; TDD in 2 studies [89, 101]; and formal specifications in 2 studies [99, 100]. There is no

mention of regression testing, integration testing, acceptance testing or performance testing in the corpus which are major areas of research [109, 110].

Tools and Platforms 4.3 RQ3

RQ3: What are the tools/ platforms developed for supporting gamification in software testing education?

General software testing accounts for the largest proportion (12 studies, 27%), including broad course interventions [23], [24], [46], [50], [47], [52], [51], [53], [54], [55] and secondary studies [15], [29], [20], [8]. The most well studied specific sub-domain is mutation testing with 9 studies (20%) most of which have been aided by CodeDefenders [10], [32], [16], [33], [17], [41], [25] Code Critters [34], [35] and EMVille [39]. The 7 studies located in Unit testing cover about 16% of the studies, with the following studies being located: UnitBrawl [49], Testable [56], Testing Robot Challenge [37], IntelliGame [38] and TDD gamification [40]. Exploratory testing appears in 4 studies [50], [47], [42], [57]; test coverage in 4 studies [18], [19], [48], [51]; software quality assurance in 4 studies [52], [58], [53], [59]; TDD in 2 studies [40], [48]; and formal specifications in 2 studies [9], [45]. The corpus does not include regression testing, integration testing, acceptance testing or performance testing, highlighting important gaps in research [20], [8].

4.3 RQ3 Tools and Platforms

RQ3: What tools and platforms have been developed to support gamification in software testing education?

Sixteen softwares that tested educational software on games were identified. The results are compared in six dimensions in Table 4.

Table 4: Comparative analysis of gamified software testing Education Tools

Tool [Refs]	Testing Sub-Domain	Game Mechanics	Deployment	Evidence Level	Key Outcome
CodeDefenders [10], [25]	Mutation Testing	Attack/Defend, Competition, Points, Leaderboard	University, multi-semester, multi-institution	Strong 120+ students, multiple replications	Consistent engagement & performance gain
GAMFLEW [18]	White-Box Testing	Serious game, Challenges, Score, Levels	University, single institution	Moderate design+eval, journal published	Improved concept understanding
CoverBot [19]	Test Coverage	Turn-based game, Points, Competition	University, single institution	Preliminary user study	Improved engagement & enjoyment
Code Critters [34], [35]	Mutation Testing (young learners)	Tower Defence, Block-based coding	School/introductory level	Moderate 2 evaluation studies	Accessible testing concept learning
TSGame [36]	Test Smells (JUnit)	Serious game, Challenge tasks, Score	University course	Preliminary position paper	Feasibility of test smell gamification
Testing Robot Challenge [37]	Unit Testing	Competition vs. automated tools	University, N=15	Preliminary small experiment	Feasibility & student motivation
IntelliGame	General	Achievements,	IDE-integrated,	Strong RCT,	Improved

[38]	Testing (Java/TS)	Feedback, Points, IDE plugin	N=174	replicability study	testing motivation & behaviour
EMVille [39]	Mutation Testing (equiv. mutants)	Motivational game mechanics	Research/expert context	Preliminary design + feasibility	Expert motivation for mutant analysis
UnitBrawl [49]	Unit Testing (Java)	Battle Royale, Competition, Rounds	University course	Preliminary design evaluation	Student engagement in unit testing
FormalZ [9], [45]	Formal Specifications	Deep gamification, Analytics, Embedded game	University (Utrecht), 2 setups	Moderate 2 experience reports	Positive attitude toward formal methods
QuiZZRep [9]	General SE / Testing	Web-based quiz tool, Gamified quizzes	University (Lisbon)	Moderate experience report	Active learning, testing knowledge
Testable [56]	Unit Testing	Gamified platform, Challenges	University, Brazil	Preliminary design study	Tool feasibility for unit test teaching
SOFTTY [43], [44]	General Testing	Physical card game, Collaboration, Competition	University, Germany	Preliminary design study	Learning testing concepts collaboratively
VR Testing Game [42]	Exploratory Testing	VR environment, Testing Tours, Defect detection	University, Bulgaria	Preliminary implementation report	Innovative engagement in testing
CI Gamification [25]	Software Testing (full course)	CI Pipeline integration, Points, Achievements	University, Passau	Moderate experience report + eval	Lightweight continuous testing engagement
Bug Hunting Game [57]	Software Testing / Bug Finding	Real executable code challenges, Web-based game	University, Netherlands (since 2020)	Moderate longitudinal since 2020	Higher enthusiasm in testing classes

4.4 RQ4: Impact on Student Engagement

RQ4: How does gamification affect students' engagement and motivation towards software testing courses?

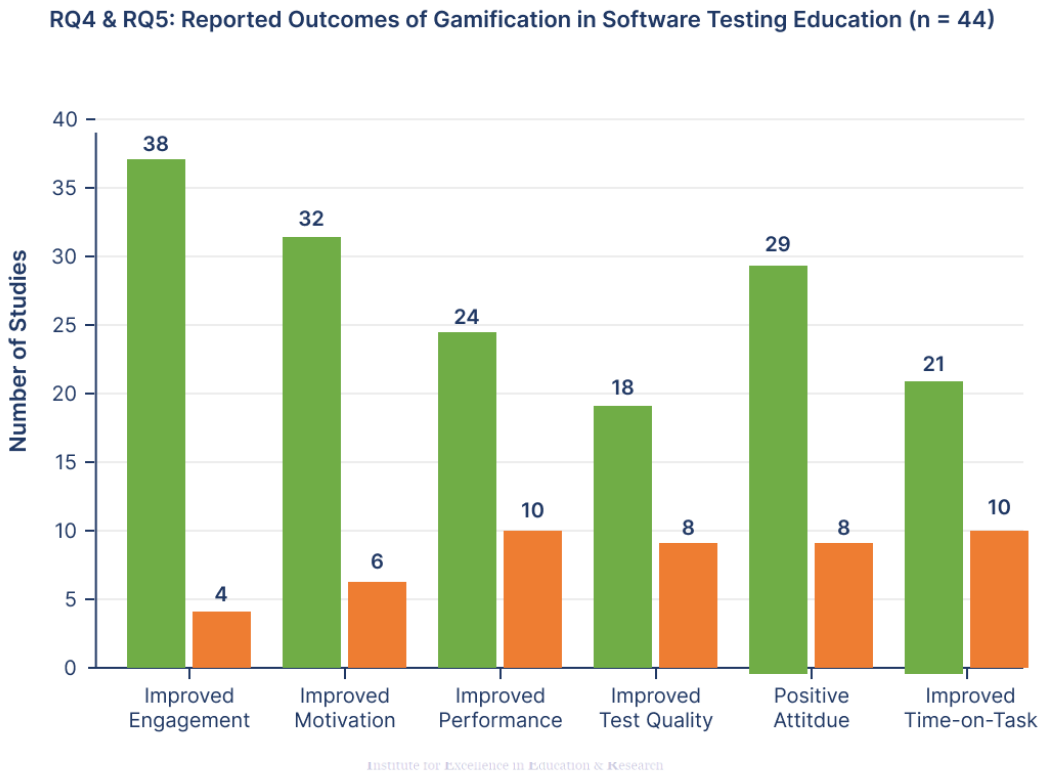


Figure 7: Reported gamification outcomes across 44 Included studies

The most common measurement of student outcomes is student engagement and motivation, with 38 of 44 studies (86%). The best evidence is from controlled experiments. Blanco et al. [23] found that there were statistically significant improvements in engagement metrics for the gamified condition. Ren [40] showed that gamification was able to significantly boost engagement and its effects remained post-gamification, as if the gamification intervention had created sustainable change in behaviours. A replicability study was conducted by Straubinger et al. [38] with 174 patients and found engagement benefits as well. According to Fraser et al. [17], [41] the Code Defenders sessions were always welcomed more than traditional labs. Straubinger and Fraser [25] identified a positive effect of CI gamification for increased test submission. This is in contrast to the study of Martins de Jesus et al. [24], [46] which found that the motivation scores were significantly higher with the gamified condition than with the control condition in several Brazilian institutions. There is much greater enthusiasm in classes, as reported by Silvis-Cividjian et al. [96] using bug hunting games since 2020. A minority show mixed types of results: Cerioli et al. [60] identified conflict between entertainment and pedagical value, Lorincz et al. [47] noticed the varying levels of engagement within game types, and Clarke et al. [53] showed that crossovers of several engagement types yielded higher results than any individual engagement type. Assessment alignment and curriculum time and technical barriers are practical barriers that synthesised

teacher perspectives highlighted by Fasolino et al. [55].

4.5 RQ5 Impact on Learning Outcomes

RQ5: How does gamification affect learning outcomes and testing performance of students?

There is a range of evidence of improvement in learning outcomes, though some is more encouraging than others. Throughout semester-long Code Defenders deployments, Fraser et al. [17], [41] report a steady improvement in mutation scores. Ren [40] showed gamification positively correlated with the improved codebase maintainability. In the gamified condition, Straubinger et al. [38] presented test coverage higher and more effective test suites. Blanco et al. [23] give the most rigorous evidence: statistically significant differences between gamified and non-gamified in both mutation scores and branch coverage and the number of tests within the test suite adequate. Martins de Jesus et al. [24], [46] are reporting similar results in knowledge acquisition between groups, but much greater persistence in performance in the gamified group. Sherif et al. [19] identified an increased understanding of test coverage concepts with the use of gamified test coverage tools; Silva et al. [18] identified an increase in understanding of white-box criteria by using GAMFLEW; Call et al. [48] report higher grades with the use of gamified SE tools. The most important evidence gap is studies on long-term skill retention, which are completely missing from the corpus [20], [8], [23].

4.6 RQ6 — Research Methodologies

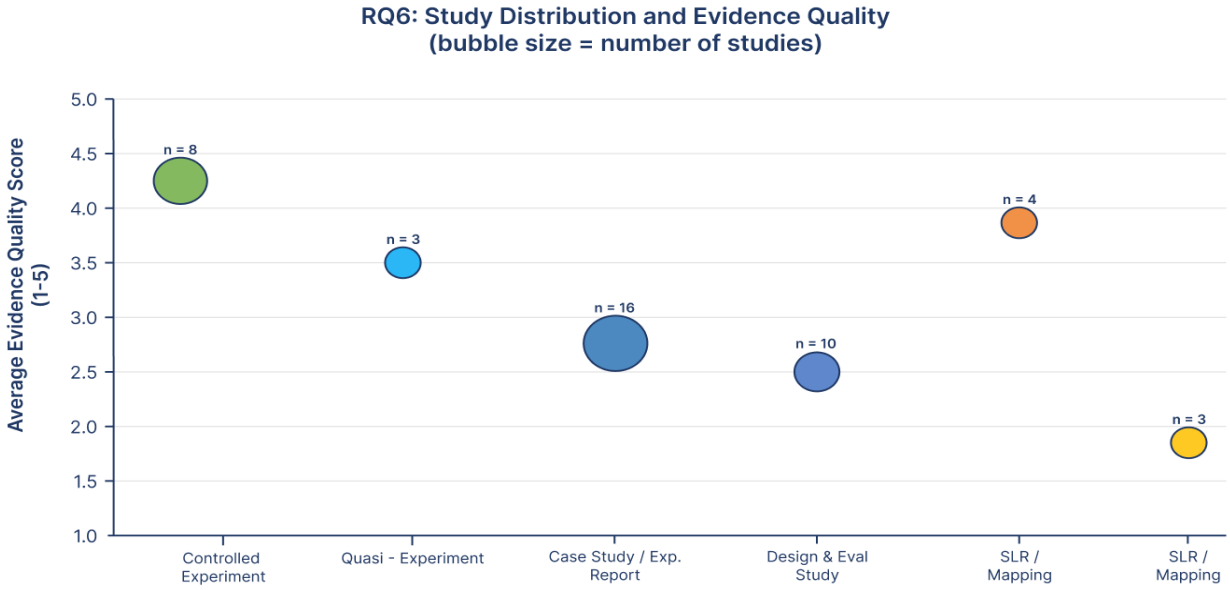


Figure 8: study design distribution and evidence quality (bubble size= number of studies).

RQ6: What kind of research methods and research designs are used?

Experience reports and case studies dominate the corpus (16 studies, 36%) [33], [17], [41], [25], [50], [47], [52], [57], [9], [45]. Design and evaluation studies account for 23% (10 studies) [18], [19], [34], [36], [37], [39], [56], [43], [44], [54]. Experiments constitute 18% (8 studies) [38], [23], [24], [46], [48], [58], [51]. Three studies [48], [51], [53] use quasi-experimental designs. Secondary studies account for 7 papers [15], [29], [21], [31], [20], [8], [55]. Most of the geographic distribution is from European (Germany, Italy, Portugal, Spain, Netherlands, Romania, Bulgaria,

Austria) and Brazilian sources, with some from the USA, Iran and other institutions. This course is designed to enable students to identify limitations and open challenges in their daily lives.

Table 5: Summary of RQ Finding Across 44 Included Studies

RQ	Key Finding	Supporting Studies	Evidence Strength
RQ1	Points (41%), competition (32%), serious games (30%), badges (27%), leaderboards (25%) dominate	[10],[25],[38],[23],[40],[43], [49]	Strong
RQ2	Mutation testing (20%), unit testing (16%); regression/integration testing absent	[10], [39], [50], [42], [15],[8]	Strong
RQ3	16 tools identified; CodeDefenders and IntelliGame most rigorously evaluated	[10],[38],[56],[42],[43],[45],[54]	Strong
RQ4	86% of studies report positive engagement; strongest evidence from controlled experiments	[17],[25],[23],[24],[46],[40],[57]	Moderate–Strong
RQ5	Improved test quality metrics in controlled studies; no longitudinal evidence	[18],[19],[38],[23],[24],[46],[40],[48]	Moderate
RQ6	Experience reports dominate (36%); controlled experiments 18%; multi-institution studies rare	[38],[23],[24],[46],[40],[48],[59]	Moderate
RQ7	Single-institution designs, no longitudinal data, game balance challenges, limited sub-domain coverage	[23],[40],[60],[20],[8],[55]	Consensus

RQ7: What are the common restrictions, hurdles and future research questions reported in the literature?

4.7.1 Methodological Limitations

The most common restriction is the short duration and single institution design [33], [17], [50], [47], [52]. There are no studies that follow children longitudinally beyond the time of the intervention [23], [40], [20]. Because of the use of heterogeneous measurement instruments across studies, quantitative comparison across studies is not possible [8], [55].

4.7.2 Design and implementation Challenges

The ability to select the appropriate code for mutation testing games is an ongoing challenge [17], [41]. Careful design is needed to balance a game for students of differing ability levels [16], [33], [23], [60]. There are practical challenges in integrating with formal assessment systems [17], [41], [25], [52].

4.7.3 Contextual Constraints

Lower-performing students can become disheartened by the use of leaderboard systems [23], [40], [47]. Finding engagement may be confounded by novelty effects [23], [24], [48]. However, as reported by teachers, there are also some measures that have a moderating effect on practical adoption, such as the lack of curriculum time and assessment alignment [55].

4.7.4 Open Research Priorities

Five open challenges are listed: (1) proof of skill retention in the use of a gamified tool [23], [40], [20]; (2) different gamification mechanics in relation to different sub-domains of the testing [20], [8]; (3) gamification for secondary education and for professional training [35], [57], [58]; (4) the adoption of AI to personalise gamification

according to individual learner profiles [58], [59]; and (5) the economic and institutional viability of the maintenance of a gamified tool [55].

5. DISCUSSION

5.1 State of Evidence

The results have confirmed that gamification in software testing education is a fruitful, developing, and mainly positive research field, which has a high level of methodological immaturity and lacks empirical evidence. There is overwhelming evidence that gamification has a positive effect on the engagement and motivation of students [61], [62], [63]. Evidence of direct improvements in learning outcomes is less strong, however, as there is a fundamental difference between the degree to which gamification can boost the amount of student involvement in testing tasks and improve the nature of the learning outcomes gained from these activities [64], [65]. A small number of controlled studies [38], [23], [40] provide the most rigorous evidence and findings that support and complement the findings of the multimedia learning theory [65] and Kolb's experiential learning theory [66]. The lack of longitudinal evidence, with no study in the corpus being able to measure whether gamification is able to generate lasting learning gains, or if it mainly effects in the short term, is the most important gap. [64], [62].

5.2 Theoretical Alignment

The game-like features are applicable to software testing. Competitive game design also takes a competitive approach, as does the adversarial logic of testing finding faults in others' code [10], [67]. In the most obvious way this is reflected in mutation testing games [32], [16] where the attack/defend dynamics are directly integrated into the game, such that fault introduction and detection are an integral part of the game. For SDT, the theories of competence through progressive challenges [18], [19], [38] autonomy through strategic choice [16], [17] and relatedness through competitive and collaborative structures [16], [47], [43] provide theoretical support. The pedagogical spectrum of tools covers Bloom's taxonomy [68] from knowledge and comprehension [18], [19] to application and analysis [16], [38] to evaluation and synthesis [36], [37] indicating that gamification can be applied in all aspects of educational objectives for software testing.

5.3 Implications for Research

Four research implications are highlighted. The first point is that there is a definite lack of controlled experiments in the literature, with only 18% compared to 59% experience reports [69], [70], [71]. To make generalisability, multi-institutional RCTs with pre-registered protocols are required [38], [69]. Second, mutation testing research is focused on a single type, mutation testing, which means that one is dependent on a single tool and research team.Second, mutation testing is very narrowly focused, with diversification needed to regression, integration, acceptance and performance testing a priority [20], [8]. Third, the potential integration of AI and gamification is an emerging trend that deserves systematic research, and personalised adaptive gamification can be considered a viable approach to tackle motivational diversity [72]. Fourth, because of the geographic concentration in European and Brazilian institutions [69], [73] it is necessary to conduct cross-cultural studies to determine transferability.

5.4 Implications for Educators

Educators are encouraged to: (1) use the most validated open source tool, CodeDefenders [16], [17], [25]; (2) incorporate multiple gamification mechanics and not solely on any single aspect [23], [24], [53], [74]; (3) design leaderboard mechanics carefully to avoid demotivating lower performers [23], [40], [47]; (4) consider using gamification as a lightweight infrastructure-compatible option, as part of the CI integration [25]; and (5) consider using physical game formats in resource-constrained contexts [47], [43], [44]. A meaningful integration of assessment into game performance is a critical success factor [25], [52] .

5.5 Threats to Validity

5.5.1 Internal Validity

Only studies found in IEEE Xplore or in Google Scholar were included in the search; papers that were only indexed in one of the other databases, such as Scopus or ACM DL or Web of Science, might not have been included [75], [76]. Grey literature and non-English publications were not included, which may have resulted in publication bias, favouring positive results [77], [78].

5.5.2 External Validity

The distribution and concentration of the evidence within the CodeDefenders research programme is predominantly European and Brazilian [69], [73] limiting the generalisability. The majority of studies were conducted on university students, and therefore, were not applicable to the secondary level of education or professional training [63], [79].

5.5.3 Construct Validity

Different types of measurement instruments used in different studies make it difficult to draw conclusions about the engagement and learning constructs across the different studies. Future research should use common tools for measuring engagement with testing and skill acquisition, such as the tools outlined in [71], [80].

5.5.4 Conclusion Validity

The narrative synthesis is a process which incorporates interpretative subjectivity [81], [82]. The small number of controlled experimental studies [38], [23], [24], [40] have a significant impact on the conclusions and more independent replications of these studies [83], [84] are welcome.

6. CONCLUSION

The appendix outlines the key findings of the study.

This systematic literature review has been able to synthesize the existing literature on gamification and software testing education on 44 primary studies published between 2015 and 2025 in a comprehensive way following the PRISMA 2020 method. The most common terms of the gamification vocabulary include points and scoring (41%), competition mechanics (32%), serious game frameworks (30%), badges (27%) and leaderboards (25%) [10], [16], [38], [23], [24]. Sixteen tools were catalogued of which CodeDefenders [16] and IntelliGame [38] were the most thoroughly evaluated. Student engagement has been shown to increase in 86% of the studies included, and statistically significant controlled experimental evidence of improvements in test quality metrics has been found in studies by Blanco et al. [23], Ren [40] and Straubinger et al. [38]. Gamification still has some uncharted waters to be explored, including regression, integration, acceptance, and performance testing [8], [20], [8]. There is no evidence at all that can be found on the literature regarding the longitudinal aspect, which is the most important one.

6.2 Original Contributions

This review offers four original contributions: (1) the first systematic literature review (SLR) that is compliant with the PRISMA guidelines with a specific focus on gamification in software testing education [75], [27]; (2) a structured taxonomy of 16 tools and evidence bases [16], [18], [34], [38], [39]; (3) a synthesised and calibrated evaluation of empirical evidence quality across 44 primary studies [23], [24], [40], [38]; and (4) a seven-point research agenda, identifying the open challenges and future directions of the field [10], [20] [8], [55]

6.3 Future Research Directions

The five priority research directions identified are: (1) research into the retention of the testing skills induced by gamification (6–12 months post-intervention) [23], [40], [64]; (2) research into generalisability through multi-institutional RCTs with pre-registered protocols [38], [69], [71]; (3) research into gamification tools for the sub-domains of testing: regression and integration, as well as testing: acceptance and testing: performance [20], [8]; (4) research into cross-cultural transferability (testing: non-European and non-Western educational contexts) [70], [63], [73]; and (5) research into AI-personalised

gamification adapting to individual learner profiles, skill levels, and motivational needs [58], [59], [72], [85].

6.4 Closing Remarks

One of the most important and underrated computing skills is software testing. The reviewed literature in this paper has shown that gamification is a promising body of evidence-based and increasingly complex repertoire of approaches to deal with the motivational issues that have been a barrier to students' engagement in software testing education in the past [10], [23], [20] , [8]. The range of tools discovered in this review, from CodeDefenders' competitive mutation testing game [16] to IntelliGame's IDE-integrated achievement system [38], from GAMFLEW's white-box serious game [18] to VR-based exploratory testing environments [42] provide educators with a growing and documented toolkit. To achieve the potential of gamification as a transformative pedagogical approach and to ensure the reliability of the software created by the next generation of engineers [1], [20], [64], [75] gaps identified for this synthesis will need to be closed through longitudinal studies, rigorous multi-institutional experiments, sub-domain diversification, and AI-personalisation.

References:

Sommerville, I., *Software engineering tenth edition*. 2016, Pearson Education Limited.

Beizer, B. and J. Wiley, *Black box testing: Techniques for functional testing of software and systems*. iEEE Software, 1996. **13**(5): p. 98.

Myers, G.J., et al., *The art of software testing*. Vol. 2. 2004: Wiley Online Library.

Bourque, P. and R.E. Fairley, *SWEBOK: guide to the software engineering body of knowledge*. 2014: IEEE Computer Society.

Tricentis. *Software Watch 5th Edition*. 2020 [cited 2026; Available from: https://www.tricentis.com/resources.

Garousi, V. and J. Zhi, *A survey of software testing practices in Canada*. Journal of Systems and Software, 2013. **86**(5): p. 1354-1376.

Kasurinen, J., O. Taipale, and K. Smolander, *Software test automation in practice: empirical observations*. Advances in Software Engineering, 2010. **2010**(1): p. 620836.

Minhas, N.M. and A.A. Barrett. *A Snapshot of Software Testing Education and Instructional Interventions*. in *2024 31st Asia-Pacific Software Engineering Conference (APSEC)*. 2024. IEEE.

Vos, T.E., et al. *Tutorial on a Gamification Toolset for Improving Engagement of Students in Software Engineering Courses*. in *2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T)*. 2020. IEEE.

Fraser, G. *Gamification of software testing*. in *2017 IEEE/ACM 12th International Workshop on Automation of Software Testing (AST)*. 2017. IEEE.

Deterding, S., et al. *From game design elements to gamefulness: defining" gamification"*. in *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*. 2011.

Hamari, J., J. Koivisto, and H. Sarsa. *Does gamification work?-a literature review of empirical studies on gamification*. in *2014 47th Hawaii international conference on system sciences*. 2014. Ieee.

Ryan, R.M. and E.L. Deci, *Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being*. American psychologist, 2000. **55**(1): p. 68.

Kapp, K.M., *The gamification of learning and instruction: game-based methods and strategies for training and education*. 2012: John Wiley & Sons.

Pedreira, O., et al., *Gamification in software engineering–A systematic mapping*. Information and software technology, 2015. **57**: p. 157-168.

Rojas, J.M., et al. *Code defenders: crowdsourcing effective tests and subtle mutants with a mutation testing game*. in *2017 IEEE/ACM*

39th International Conference on Software Engineering (ICSE). 2017. IEEE.

Fraser, G., et al. *Gamifying a software testing course with code defenders*. in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. 2019.

Silva, M., A.C. Paiva, and A. Mendes, *GAMFLEW: serious game to teach white-box testing*. Software Quality Journal, 2025. **33**(1): p. 5.

Sherif, E., et al. *Gamification to aid the learning of test coverage concepts*. in *2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T)*. 2020. IEEE.

Fulcini, T., et al., *A review on tools, mechanics, benefits, and challenges of gamified software testing*. ACM Computing Surveys, 2023. **55**(14s): p. 1-37.

Di Nardo, V., et al., *Usage of gamification techniques in software engineering education and training: A systematic review*. Computers, 2024. **13**(8): p. 196.

Herculani, J., *Gamification in Software Engineering Education: a Tertiary Study*. Conference, 2023.

Blanco, R., et al., *Can gamification help in software testing education? Findings from an empirical study*. Journal of Systems and Software, 2023. **200**: p. 111647.

de Jesus, G.M., et al., *Is it worth using gamification on software testing education? An extended experience report in the context of undergraduate students*. Journal of Software Engineering Research and Development, 2020. **8**: p. 6: 1-6: 19.

Straubinger, P. and G. Fraser. *Gamifying a software testing course with continuous integration*. in *Proceedings of the 46th International Conference on Software Engineering: Software Engineering Education and Training*. 2024.

Page, M.J., et al., *The PRISMA 2020 statement: an updated guideline for reporting systematic reviews*. Revista panamericana de salud publica, 2022. **46**: p. e112.

Kitchenham, B. and S. Charters, *Guidelines for performing systematic literature reviews in software engineering*. 2007.

Malone, T.W., *Toward a theory of intrinsically motivating instruction*. Cognitive science, 1981. **5**(4): p. 333-369.

Souza, M.R.D.A., et al. *Games for learning: bridging game-related education methods to software engineering knowledge areas*. in *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET)*. 2017. IEEE.

Al-Adwan, A., et al., *Detection of deepfake media using a hybrid CNN–RNN model and particle swarm optimization (PSO) algorithm*. Computers, 2024. **13**(4): p. 99.

Tonhão, S., et al. *Gamification in software engineering education: A tertiary study*. in *Proceedings of the XXXVII Brazilian Symposium on Software Engineering*. 2023.

Rojas, J.M. and G. Fraser. *Code defenders: a mutation testing game*. in *2016 IEEE Ninth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. 2016. IEEE.

Fraser, G., A. Gambi, and J.M. Rojas. *A preliminary report on gamifying a software testing course with the code defenders testing game*. in *Proceedings of the 3rd European Conference of Software Engineering Education*. 2018.

Straubinger, P., L. Caspari, and G. Fraser. *Code critters: A block-based testing game*. in *2023 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. 2023. IEEE.

Straubinger, P., L. Bloch, and G. Fraser. *Engaging young learners with testing using the code critters mutation game*. in *2024 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. 2024. IEEE.

Fasolino, A.R. and P. Tramontana. *Test smells learning by a gamification approach*. in *Proceedings of the 3rd ACM International Workshop on Gamification in Software*

Development, Verification, and Validation. 2024.

Fasolino, A.R., C.M. Accetto, and P. Tramontana. *Testing robot challenge: A serious game for testing learning*. in *Proceedings of the 3rd ACM International Workshop on Gamification in Software Development, Verification, and Validation*. 2024.

Straubinger, P., et al., *Gamifying Testing in IntelliJ: A Replicability Study*. Proceedings of the ACM on Software Engineering, 2025. **2**(ISSTA): p. 2407-2429.

Houshmand, M. and S. Paydar. *Emville: A gamification-based approach to address the equivalent mutant problem*. in *2017 7th International Conference on Computer and Knowledge Engineering (ICCKE)*. 2017. IEEE.

Ren, W. *Gamification in test-driven development practice*. in *Proceedings of the 2nd International Workshop on Gamification in Software Development, Verification, and Validation*. 2023.

Fraser, G., A. Gambi, and J.M. Rojas. *Teaching software testing with the code defenders testing game: Experiences and improvements*. in *2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. 2020. IEEE.

Nikolova, T., et al., *Software Testing as Gamification in Education via VR Games*.

Soska, A., J. Mottok, and C. Wolff, *An Experimental Card Game for Software Testing*. 2017.

Soska, A., J. Mottok, and C. Wolff. *Pattern oriented card game development: SOFTTY–A card game for academic learning of software testing*. in *2017 IEEE Global Engineering Education Conference (EDUCON)*. 2017. IEEE.

Prasetya, I., et al. *Can Learning Formal Specification Be Fun?–Experience and Perspective*. in *2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. 2020. IEEE.

de Jesus, G.M., et al. *Is it worth using gamification on software testing education? An experience report*. in *Proceedings of the XVIII Brazilian Symposium on Software Quality*. 2019.

Lőrincz, B., B. Iudean, and A. Vescan. *Experience report on teaching testing through gamification*. in *Proceedings of the 3rd International Workshop on Education through Advanced Software Engineering and Artificial Intelligence*. 2021.

Call, M.W., E. Fox, and G. Sprint, *Gamifying software engineering tools to motivate computer science students to start and finish programming assignments earlier*. IEEE Transactions on Education, 2021. **64**(4): p. 423-431.

Materazzo, A., et al. *Survival of the tested: Gamified unit testing inspired by battle royale*. in *2023 IEEE/ACM 7th International Workshop on Games and Software Engineering (GAS)*. 2023. IEEE.

Costa, I.E.F. and S.R.B. Oliveira. *The use of gamification to support the teaching-learning of software exploratory testing: an experience report based on the application of a framework*. in *2020 IEEE Frontiers in Education Conference (FIE)*. 2020. IEEE.

Ramasamy, V., et al. *A minimally disruptive approach of integrating testing into computer programming courses*. in *Proceedings of the 2nd International Workshop on Software Engineering Education for Millennials*. 2018.

Moser, G., et al. *Teaching software quality assurance with gamification and continuous feedback techniques*. in *2021 IEEE Global Engineering Education Conference (EDUCON)*. 2021. IEEE.

Clarke, P.J., et al., *Combining learning and engagement strategies in a software testing learning environment*. ACM Transactions on Computing Education (TOCE), 2021. **22**(2): p. 1-25.

Fu, Y. and P.J. Clarke. *Gamification-based cyber-enabled learning environment of software testing*. in *2016 ASEE Annual Conference & Exposition*. 2016.

IEER Institute for Excellence in Education & Research

Fasolino, A.R., et al., *Teachers' Perspective on Software Testing Education*. ACM Transactions on Computing Education, 2025. **26**(1): p. 1-36.

Marabesi, M. and I.F. Silveira. *Towards a gamified tool to improve unit test teaching*. in *2019 XIV latin american conference on learning technologies (LACLO)*. 2019. IEEE.

Silvis-Cividjian, N., et al. *Bug hunting games to add enthusiasm in software testing and programming classes*. in *2025 IEEE/ACM 37th International Conference on Software Engineering Education and Training (CSEE&T)*. 2025. IEEE.

Ödevci, B.B., et al. *A Machine Learning Model for Predicting Performance of Gamified Software Test Specialist*. in *2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*. 2022. IEEE.

Velázquez-García, L., A. Cedillo-Hernández, and I. Domínguez-Jiménez. *AI-Assisted Gamified Experiences for Teaching IEEE Standards in Software Engineering Education*. in *2025 International Conference on Education Technology and Computers (ICETC)*. 2025. IEEE.

Cerioli, M., M. Leotta, and M. Ribaudo. *Enlisting students in gamifying software testing education: Results and key insights*. in *Proceedings of the 3rd ACM International Workshop on Gamification in Software Development, Verification, and Validation*. 2024.

Connolly, T.M., et al., *A systematic literature review of empirical evidence on computer games and serious games*. Computers & education, 2012. **59**(2): p. 661-686.

Seaborn, K. and D.I. Fels, *Gamification in theory and action: A survey*. International Journal of human-computer studies, 2015. **74**: p. 14-31.

Dicheva, D., et al., *Gamification in education: A systematic mapping study*. Journal of educational technology & society, 2015. **18**(3): p. 75.

Deci, E.L., R. Koestner, and R.M. Ryan, *A meta-analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation*. Psychological bulletin, 1999. **125**(6): p. 627.

Silva, A.C.d., *MAYER, Richard E. Multimedia Learning, New York: Cambridge University Press, 2009*. Ensaio Pesquisa em Educação em Ciências, 2017. **19**: p. 619-622.

Kolb, D.A., *Experiential learning: Experience as the source of learning and development*. 2014: FT press.

McGonigal, J., *Reality is broken: Why games make us better and how they can change the world*. 2011: Penguin.

Bloom, B.S., *Taxonomy of.* Educational Objectives, 1956. **250**(2025): p. 3.

Wohlin, C., et al., *Experimentation in software engineering*. Vol. 236. 2012: Springer.

Creswell, J.W. and J.D. Creswell, *Research design: Qualitative, quantitative, and mixed methods approaches*. 2017: Sage publications.

Kitchenham, B.A., et al., *Preliminary guidelines for empirical research in software engineering*. IEEE Transactions on software engineering, 2002. **28**(8): p. 721-734.

Shute, V.J., *Focus on formative feedback*. Review of educational research, 2008. **78**(1): p. 153-189.

Stol, K.-J. and B. Fitzgerald, *The ABC of software engineering research*. ACM Transactions on Software Engineering and Methodology (TOSEM), 2018. **27**(3): p. 1-51.

Zichermann, G. and C. Cunningham, *Gamification by design: Implementing game mechanics in web and mobile apps*. 2011: " O'Reilly Media, Inc.".

Page, M.J., et al., *The PRISMA 2020 statement: an updated guideline for reporting systematic reviews*. bmj, 2021. **372**.

Petticrew, M. and H. Roberts, *Systematic reviews in the social sciences: A practical guide*. 2008: John Wiley & Sons.

Moher, D., et al., *Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement*. Bmj, 2009. **339**.

IEER
Institute for Excellence in Education & Research

Gough, D. and M. Richardson, *Systematic reviews*, in *Advanced research methods for applied psychology*. 2024, Routledge. p. 71-86.

Caponetto, I., J. Earp, and M. Ott. *Gamification and education: A literature review*. in *European conference on games based learning*. 2014.

Caldiera, V.R.B.-G. and H.D. Rombach, *Goal question metric paradigm*. Encyclopedia of software engineering, 1994. **1**(528-532): p. 6.

Oliver, S. and J. Tripney, *Systematic review and meta-analysis*. The BERA/SAGE Handbook of Educational Research, Londres: Sage, 2017: p. 452-475.

Tranfield, D., D. Denyer, and P. Smart, *Towards a methodology for developing evidence-informed management knowledge by means of systematic review*. British journal of management, 2003. **14**(3): p. 207-222.

Glass, R.L., I. Vessey, and V. Ramesh, *Research in software engineering: an analysis of the literature*. Information and Software technology, 2002. **44**(8): p. 491-506.

Hannay, J.E., et al., *The effectiveness of pair programming: A meta-analysis*. Information and software technology, 2009. **51**(7): p. 1110-1122.

Wieringa, R., *Design science methodology for information systems and software engineering*. 2014: Springer.

IEER
Institute for Excellence in Education & Research