Link-Traced Amplification: How Telegram Redistributes YouTube Political Content at Scale

Software Quality Journal (2026) 34:49
https://doi.org/10.1007/s11219-026-09781-4

RESEARCH

GAppium experiment: Applying gamification to mobile GUI testing to increase effectiveness, efficiency, engagement

Lorenzo Laudadio[1] · **Riccardo Coppola**[1] · **Marco Torchiano**[1]

Received: 5 December 2025 / Accepted: 1 August 2026

Abstract
Testing of mobile applications is a crucial activity that aims to reveal bugs and defects affecting software running on our smartphones. Usually, mobile applications feature complex Graphical User Interfaces (GUIs) as the only interface with the human user, therefore proper End-to-End testing of mobile applications should also include GUI testing. In recent years, several automated testing methods have been proposed for mobile GUI testing; however, due to the dynamic and interactive nature of GUIs, they often fail to create realistic use cases. Exploratory testing is a popular GUI testing technique that involves testers freely exploring the application under test. This technique heavily relies upon testers' experience and creativity, and is negatively influenced by human factors like stress and boredom, which are likely to appear in unmotivated testers. Gamification, i.e., the introduction of game design mechanics to tasks of other nature, has recently gained much popularity in business contexts, being applied to a wide range of software-related activities. In this article, we propose a gamification approach to mobile Exploratory Testing to improve the effectiveness, efficiency, and engagement of the software testers. We measure the impact of gamification with an in-vivo experiment involving students from a master's degree software engineering course, adopting a fully-randomized experiment design. On the efficiency side, results show a positive impact of gamification on the explored pages – a measure of effectiveness – and on the explored pages per unit of time – which we adopted as an efficiency measure. Moreover, a final survey presented to students reveals that most of them believe that it is a good idea to use gamification in Exploratory Testing, and that they would use gamification in future if they had to do Exploratory Testing. Finally, the majority of participants report having fun while using the system. Our results highlight that gamification features can have a significant impact if introduced in a classical testing environment, possibly increasing effectiveness, efficiency and perceived engagement of the testers. We believe this research could serve as a starting point for researchers approaching the mobile testing environment, as well as industry practitioners working in the field.

Keywords Gamification · Software · Testing · End-to-End testing · GUI testing

Extended author information available on the last page of the article

Published online: 31 August 2026

1 Introduction

Software testing is an essential part of the software development process. The goal of software testing is to find as many defects as possible in the Application Under Test (AUT), before it reaches the end users. End-to-End (E2E) Testing focuses on testing the application as a unique thing, adopting the end users' perspective. In web and mobile applications, this includes testing through the main users' interaction method: the Graphical User Interface (GUI).

Several methods have been proposed for GUI testing, which can be divided into two main categories: methods involving human testers and automated methods. While automated testing methods are able to produce test cases fast and can reach good coverage of the AUT, they may fail at generating realistic use cases (Evers et al., 2020; Wang et al., 2017). Moreover, test cases generated through automatic methods are often hard to maintain (Shamshiri et al., 2018). Previous researchers have shown that manual methods are still very popular among mobile developers and testers (Mahmud et al., 2025). A widely adopted technique involving human testers is Exploratory Testing (ET) (Bach, 2019), an interactive testing process which relies upon testers' experience, creativity and intuition to create realistic use cases. With ET, test cases are not defined in advance. Instead, they are defined and executed on the fly, while testers are free to explore the AUT with no particular constraints.

Surveys conducted on software testers have shown that software testing in general is perceived as a boring and time-consuming activity (Straubinger & Fraser, 2023). Human factors like boredom, fatigue and stress can have a negative impact on performance in manual testing activities.

We propose gamification, defined as the application of game design mechanics to non-ludic contexts (Deterding et al., 2011), as a technique to overcome this limitation. Gamification has recently become very popular, especially in business environments, being applied to several software engineering-related activities, including software testing as well. Gamification is used to turn repetitive, boring tasks into appealing and entertaining ones, thus increasing the engagement level of human beings.

In the present paper, we present a novel approach to the testing of mobile GUI-based applications by enriching the ET activity with gamification. Our main goal is to improve the testers' performances, focusing on effectiveness, efficiency and engagement.

Our research method consists of a randomized experiment where participants had to perform exploratory GUI testing on two different applications. The testing tool has been modified to include typical gamification features, like leaderboards, badges, customization and progress bars indicating the current page coverage and incremental coverage (defined in terms of widgets)[1]. One of the two testing sessions includes gamification, while the other is Vanilla (i.e. the original Appium Inspector, without any gamified additions).

The paper is structured as follows: in Section 2 we present an overview of the current State Of The Art, showing previous results which serve as motivation for our research. In Section 3 we delve into our method: we formally define our research questions, explain our experimental design, identify the measured variables and present our analysis method. In Section 4 we present our main findings, both from a quantitative and a qualitative perspective. In Section 5 we discuss the main implications of our results. Finally, in Section 6 we conclude our paper with some final considerations about potential future work.

[1] The testing tool is publicly available on Zenodo at the following link: https://doi.org/10.5281/zenodo.14781340

2 Background

Exploratory Testing (ET) was defined by James Bach as "simultaneous learning, test design, and test execution" (Bach, 2019). This powerful creativity-based approach to software testing has been extensively explored in the literature (Gebizli & Sözer, 2017; Mårtensson et al., 2021; Leveau et al., 2022). One of the main advantages of ET is the fact that it does not require a formal definition of test cases, which are instead generated and executed on-the-fly. Creativity, experience and domain knowledge are central. ET puts an emphasis on testers' freedom, autonomy and learning, by letting them experiment with the Application Under Test (AUT) (Costa & Oliveira, 2021). ET tools is very often used in combination with automated testing tools (Pfahl et al., 2014) or scripted testing (Ghazi et al., 2018), to improve the quality of test cases. By its nature, ET can easily be adapted to dynamic Agile frameworks, like SCRUM, where it is considered 'highly effective' or 'effective' by practitioners (Neri et al., 2025).

The cost required by manual testing can be mitigated with Capture & Replay (C&R) tools, i.e. tools that allow testers to record their actions during the testing session to replicate them later (Di Martino et al., 2021). Tools like SikuliX and WebRR fall under this category (RaiMan's SikuliX, 2025; Long et al., 2020). Some effort has been put into creating C&R tools from existing testing frameworks. For instance, Espresso Test Recorder that exploits the Espresso Testing Framework API to record and replicate user interactions on the AUT from within Android Studio (Negara et al., 2019).

A further advancement in this field has been made with Augmented Testing (AT). AT consists of displaying an additional visual layer on top of the AUT that provides useful data to help testers make decisions during their ET session. AT has also been the object of an industrial evaluation, with positive outcomes (Nass et al., 2019).

However, these tools do not address the main issue related to manual testing: even if we can record tester's actions, the testers still have to perform those actions manually. If they are bored or tired, the whole performance will be negatively impacted (Deak et al., 2016).

Gamification consists of applying game-design elements to diverse types of tasks to increase motivation and possibly improve performance. Several frameworks have been developed, encompassing the main gamification techniques. One of the most popular and widely adopted in industry is the Octalysis Framework by Chou (2015), which identifies 8 typical game elements called Core Drives. Gamification has been applied to all phases of the software development process, including software testing, with systematic mappings documenting its adoption across development, testing and education (Pedreira et al., 2015; Porto et al., 2021). Most primary studies focus on the educational field; the majority report positive effects (Ren, 2023; Fasolino et al., 2024). In particular, Blanco et al. conducted a controlled experiment comparing gamified and non-gamified testing courses (Blanco et al., 2023). Their results show that gamified students were more engaged and achieved better results.. Fraser et al. introduced Code Defenders – a mutation analysis game – in a university-level software testing course (Fraser et al., 2019). They reported an overall improvement in the students throughout the course, with good engagement and satisfaction levels. Beyond education, gamification has been applied to unit testing through CI tools. Straubinger and Fraser developed Gamekins, a Jenkins plugin that motivates developers to write more and better tests by incorporating challenges, quests, leaderboards, and achievements tied to commits. In a subsequent study, they explored whether gamification could

also support exploratory unit testing directly within the IDE with an in-vivo experiment, with participants gathering nearly 90% line coverage and detecting most of the bugs (11 out of 14) (Straubinger & Fraser, 2024). Valle et al. applied the concept of gamification in an industrial training environment (Valle et al., 2021), highlighting positive outcomes on the engagement of testers and the quality of test cases. Regarding the possible benefits of gamification on manual exploratory software testing, the most relevant related work has been made by Coppola et al. (2024). This work provides the most complete empirical study on gamified exploratory GUI testing to date. They augmented a Capture & Replay web application testing tool with a leaderboard, a progress bar and exploration highlights, comparing the original and gamified versions in a controlled experiment with 144 participants, measuring effectiveness, efficiency, test cases realism and user experience. Results are mixed, but confirm that gamification can have a tangible effect on how testers create test cases, thus warranting further research. Our work differs along three key dimensions: first, we target mobile applications via the Appium framework, and touch-specific interaction paradigms; second, our gamification feature set explicitly covers autonomy and identity dimensions grounded in Self-Determination Theory (avatar, username customization) (Deci & Ryan, 1985) (that we will introduce and expand in Section 3), and third, our experiment involves 238 participants, providing greater statistical power and generalizability than the previous studies, and extends the empirical evidence to a new platform domain where no prior gamification study exists. Recently, Fulcini et al. performed a multivocal literature review on the existing relationship between gamification and software testing (Fulcini et al., 2023), observing a majority of positive opinions on gamification from the literature, especially on the user experience side. These results encourage further investigation into the topic.

3 Method

The goal of this study is to investigate the effects of gamification on exploratory testing of mobile applications from three different perspectives: effectiveness, efficiency, and engagement. Our approach in this sense is guided by the Goal Question Metric framework, by Basili and Weiss (1984).

3.1 Goal and research questions

Table 1 presents our goal definition scheme, based on the GQM template. The GQM template allows to define research questions starting from high level goals. In particular, the research questions are derived from the Action "With respect to". The research questions we derived for the experiment are the following:

- **RQ1:** Does the adoption of gamification mechanics have a positive impact on the effectiveness of mobile exploratory testing?
- **RQ2:** Does the adoption of gamification mechanics have a positive impact on the efficiency of mobile exploratory testing?
- **RQ3:** Does the adoption of gamification mechanics have a positive impact on the perceived engagement of mobile exploratory software testers?

Table 1 Formulation of the goals of the paper according to the Goal-Question-Metric template

Action	Object
Analyse	The effects of gamification on ET
For the purpose of	Evaluation
With respect to their	Effectiveness, efficiency, engagement level
From the point of view of	Testers
In the context of	Software Engineering course

3.2 Experimental setup

The study aims to investigate the impact of gamification on exploratory testing with an in-vivo experiment. The experiment was designed as a within-subjects 2×2 full factorial. Participants had to test two different existing Android applications in which some bugs had been injected. Participants were not aware of the bug injection; they were just told to seek possible bugs and report them. The choice of the applications to be tested was based on the following criteria:

1. The applications must be open-source, to allow us to perform bug injection.
2. The applications should be under active development (last commit <2 years).
3. The applications should have a sufficient level of explorability.
4. The applications should be known for other experiments in the literature.

Regarding number 3 criteria, during a preliminary evaluation of the tool, it became clear that the depth of exploration rarely exceeded three pages; for this reason, a study-specific heuristic was used, even though there is no established definition in the literature for the level of explorability. Conversely, applications with only 1–2 web pages are considered extremely limiting for capturing interactions. Therefore, we judge a 'sufficient level of explorability' as having at least three pages where the contained widgets change and the functions provided by the pages are different. This allows us to exclude single-page applications (like, for example, an audio recorder). Typically, an application with a main landing page, a settings page and an edit content page will just suffice.

As a result, we chose two popular open-source applications taken from the F-Droid mobile catalogue (Fdroid, n.d.): OmniNotes (federicoiosue, n.d.) – a note-taking app – and PassAndroid (ligi, 2025) – an app that manages tickets, coupons, loyalty cards, boarding passes, etc. Both applications have been adopted in other experiments (Mikhaltsov & Sorokin, 2022; Mukherjee et al., 2022; Coppola et al., 2020; Romdhana et al., 2022). Table 2 shows the complexity of the AUTs. For the complexity analysis we used Lizard, a widely known and open-source static analysis tool (terryyin, n.d.), and shellscript-based parsing (to get the number of activities and fragments). As it can be seen, OmniNotes is a significantly larger application, with approximately six times the non-commented lines of code (17188 vs 2817), and nearly seven times the number of functions (1347 vs 197), indicating substan-

Table 2 Complexity of the two AUTs

AUT	nloc	avg nloc	avg CCN	#functions	#activities	#fragments
OmniNotes	17188	9.6	2.2	1347	12	14
PassAndroid	2817	8.3	2.6	197	13	58

tially greater size and functional scope. PassAndroid, despite the smaller codebase, exhibits a richer UI structure, with 59 fragments compared to 14 in OmniNotes, suggesting a more complex architecture and a higher number of distinct UI states that testers may encounter. Both apps show comparable and relatively low average cyclomatic complexity (CCN) (2.2 and 2.6 respectively), below the threshold of 10, commonly considered indicative of high complexity (McCabe, 1976) and similar average function lengths (9.6 and 8.3 nloc).

The tasks for the participants in the experiment consisted of testing the applications using two distinct tools. We provided the student with two different versions of the Appium Inspector testing tool (Welcome - Appium Inspector, n.d.): the Vanilla (unmodified) version and the Gamified version (GAppium Laudadio et al., 2025); in this latter version, we implemented a few typical gamification mechanics: Coverage Progress Bars, Badges, Customization and Leaderboards. These elements are depicted and highlighted with different colors in Fig. 1.

The features were selected based on the Self-Determination Theory (SDT) (Deci & Ryan, 1985). SDT suggests that motivation increases when three psychological needs – competence, autonomy, and relatedness – are satisfied. Progress bars and badges target competence by providing feedback on coverage and recognizing achievement; customization supports autonomy by fostering tool ownership; leaderboards address relatedness through social comparison. These motivational effects are expected to translate into measurable performance outcomes: progress bars are supposed to improve effectiveness and efficiency by guiding test coverage; badges and leaderboards are supposed to drive effectiveness by incentivizing thoroughness; customization and leaderboards are supposed to sustain engagement over time. Table 3 summarizes the features, their motivational effect, the performances they are supposed to target and the Octalysis Framework related Core Drives.

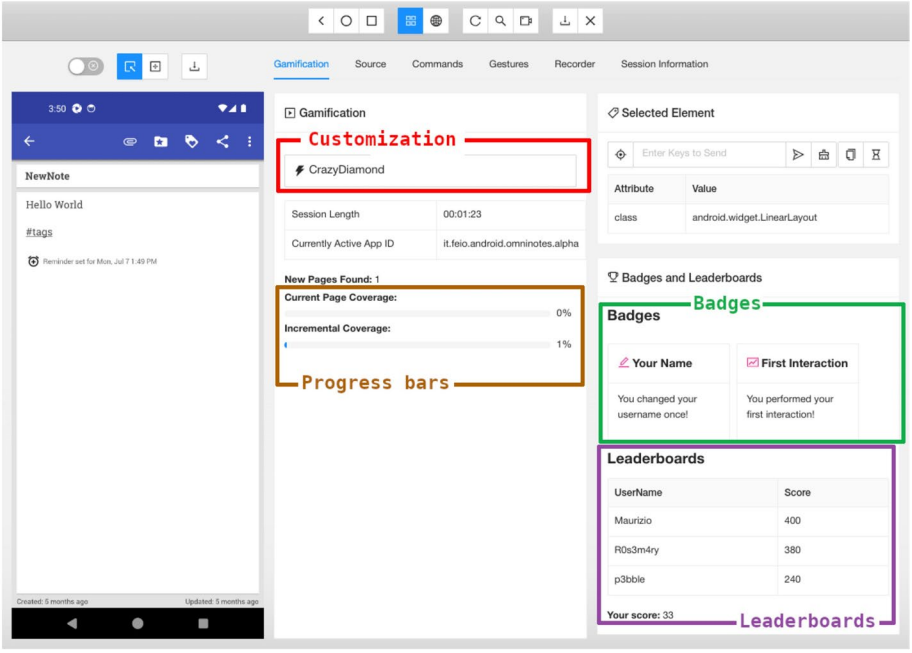

Fig. 1 A typical GAppium session

As it can be seen, the implemented gamification features are related to Development & Accomplishment, which consists in making progress and achieving mastery, and Ownership & Possession, which is triggered from owning and improving things. We have decided to implement these features because we deemed them to be the most suitable for this kind of platform, although work is currently underway to implement additional ones.

It is worth to mention that the leaderboard is just simulated, i.e. it is not synchronized among all the players, since the testing platform does not support any online multiplayer feature currently. However, since it is not being updated in real time, there is no difference between a fake leaderboard and a real one. A real leaderboard would only have a visible impact if it was updated in real time during the sessions. "Fake" or "simulated" leaderboards have been used in the past and are still used nowadays in experimental settings as placebo effects (Heasman & Gillespie, 2019; Stoeve et al., 2022). Moreover, research has demonstrated that in gamification, human behavioral responses are driven primarily by psychological expectations and perceptions rather than objective reality (Brantley et al., 2021; Winkler & Hermann, 2019). In conclusion, we do not believe that the usage of a fake leaderboard could have had a detrimental effect on the experiment outcome.

Participants were randomly divided into four groups of the same size. All groups were exposed to both treatments in sequence. The order of execution for each group is depicted in Fig. 2.

The study was conducted on 238 volunteer students from Politecnico di Torino. The student attended the software engineering course – where software testing is part of the syllabus – of the MSc in Computer Engineering. Among the participants, 167 were physically present in the experiment classroom, while 71 attended the experiment session remotely. Students taking part in the experiment were given an additional point on the final course grade. Apart from the university ID of the students, no other personal data possibly leading to unique identification of the subjects was recorded in the artefacts. An informed consent statement was available in the application form for the experiment. A preliminary analysis of the demographics shows that only 31 students had more than 1 year of experience in mobile application development, only 49 had more than 1 year of experience in software testing, and only 3 had previously performed ET.

The participants were strongly encouraged to perform sessions lasting at least 15 minutes, with a maximum time of 30 minutes per session. These limits were chosen to have a balance between the freedom of the participants and external time constraints deriving from the classroom availability, and they are in line with durations adopted for similar experi-

Table 3 The psychological components satisifed by the implemented gamification features, the performance outcome they are supposed to target, and the Octalysis framework related Core Drives.

Feature	Motivational Effect	Performance	Core Drive
Progress bars	Competence	Effectiveness, Efficiency	CD2 - Development & Accomplishment
Badges	Competence	Effectiveness	CD2 - Development & Accomplishment
Leaderboards	Relatedness	Effectiveness, Engagement	CD2 - Development & Accomplishment
Customization	Autonomy	Engagement	CD4 - Ownership & Possession

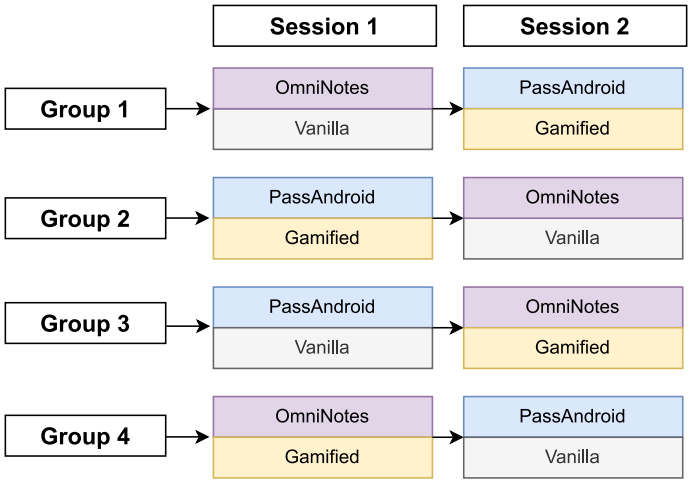

Fig. 2 Order of execution for the groups

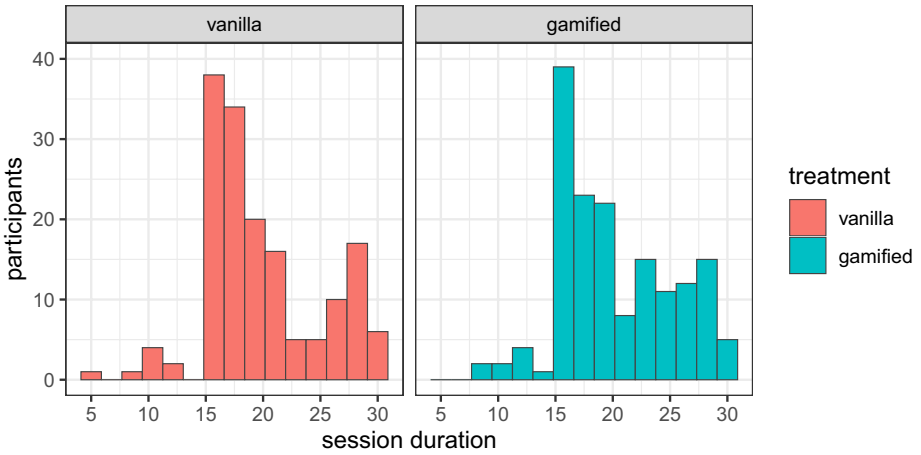

Fig. 3 Time distribution of the duration of the sessions among the participants

ments (Coppola et al., 2024; Straubinger et al., 2025; Garaccione et al., 2024). As it can be seen from Fig. 3 the vast majority of the participants respected these soft constraints. Most of them, however, closed their sessions immediately after 15 minutes.

During the experiment, a session report containing the data required for the analysis was automatically recorded. Meanwhile, students had to fill in a spreadsheet form reporting all the bugs they had found, with a short description and the time of finding. At the end of the session, participants were asked to answer a Likert questionnaire (Wikipedia, 2025) containing questions aimed at measuring the engagement and satisfaction level, as well as general usability questions adapted from the System Usability Scale (SUS) (Brooke, 1996) and general demographic questions. Students were asked to upload these artefacts to a private online university workspace at the end of the experiment.

3.3 Variables

Our experiment defines four independent variables. Treatment is the main factor, i.e. the application of gamification to the session, while Sequence, Order and AUT are co-factors. Sequence relates to the time of execution of the session: its value is 1 if the session is the first, 2 if the session is the second, while Order represents the order in which the treatment is applied to the participants.

Considering the software testing aspect we are evaluating, we identified the following dependent variables:

1. **Effectiveness:** The ISO 9001 standard defines Effectiveness as the "Extent to which planned activities are realised and planned results are achieved." (Standard, International Organization for Standardization, 2015). Exploratory Testing effectiveness reflects the ability to thoroughly explore the AUT and accurately identify defects. Following prior work on gamified GUI testing, in our experiment, we operationalize effectiveness through Explored Pages and Bug Report Precision/Recall. Explored pages – and similar metrics – are typically adopted metrics related to the effectiveness of software testing (Bures et al., 2018; Leveau et al., 2020). In white-box testing, we typically define coverage in terms of code. In our case, testers do not have access to the code of the applications, so we use the number of visited pages as a measure of exploration. To have a definition of exploration compatible with web applications, we define a *page* as a specific ordered set of widgets. In fact, the data structure representing a page is a concatenation of the names of the widgets present on the screen at a given time. It follows that, in our experiment, the number of explored pages will be equal to the number of unique strings representing the concatenations of widgets that the testers are encountering in their paths. Precision and Recall are also common metrics in data analysis, often used jointly. Bug Report Precision is defined as the number of True Positive bugs (TPs) reported divided by the number of all the bugs reported. Recall is defined as the number of TPs over the total number of bugs in the AUT.
2. **Efficiency:** Defined by ISO 9001 as the "Relationship between the result achieved and the resources used". On this side, we define three dependent variables: explored pages per unit of time, number of TPs per unit of time, and Precision/Recall per unit of time.
3. **Engagement:** to assess the engagement of testers during their testing activities, we defined a set of questions with answers on a Likert scale. Some questions are more related to the overall User eXperience (UX) and usability of the system, and taken from a classical System Usability Scale (SUS) questionnaire[2], while others are more specific to the engagement. Engagement and Usability were measured using non-overlapping questionnaire items, targeting affective/motivational responses and perceived ease-of-use, respectively.

The considered variables are reported in Table 4.

[2] It is worth noting that usability and engagement may not be fully independent constructs in this context. We assume that a system perceived as easy to use and consistent may lower cognitive load, thereby freeing attentional resources that can be redirected toward engagement with the gamification elements. Conversely, high perceived complexity or frustration may suppress engagement regardless of the gamification design. This potential mediation effect – where usability acts as a prerequisite for engagement – is however an assumption of this study and was not formally tested; we acknowledge it as a direction for future work.

3.4 Testing environment

The testing environment was composed of the following components:

- The two versions of the Appium Inspector testing tool (respectively with and without gamification)
- The two AUTs: PassAndroid and OmniNotes.

Participants received instructions on how to set up the environment for testing on their personal computers 10 days before the actual experiment. On the experiment day, students received live instructions on how to run the testing applications and to run the testing activity. Before the experiment trials, a 30-minutes lecture about Exploratory Testing was provided. The students were taught the basics of scenario-based testing and the way manual exploration of the GUI of a webapp can be conducted to follow the main routes that a user can perform. Examples were provided to the students about how to detect a bug in a sample application, and how to create a bug report within the spreadsheet they were provided with. Remote students received a video tutorial with the same content.

Before starting the bug injection process, we identified 4 bugs which were already present in the PassAndroid application. The injected bugs belong to the three classes of mutants defined by Alegroth et al. in previous works (Alegroth et al., 2015): Remove Listener, Duplicate/Insert and Modify. Table 5 lists the bugs injected in both AUTs. Bug IDs marked with an asterisk (*) were already present before our injection process.

At the end of the bug injection process, we wanted both the applications have the same number of known bugs, including the ones which were already present before the injection.

Table 4 The considered variables

Name	Description	Values	Indep./Dep.
Treatment	The type of treatment applied to the subject	Gamified, Vanilla	I
Sequence	The sequence number of the session (1 if it is the first session, 2 if it is the second)	1, 2	I
Order	The order in which treatments are administered to the subject	Gamified-Vanilla, Vanilla-Gamified	I
AUT	The application which is being tested in this session	PassAndroid, OmniNotes	I
Bug Report Precision	The bug report precision value obtained by a tester during the session	Real numbers	D
Bug Report Recall	The bug report recall value obtained by a tester during the session	Real numbers	D
Explored pages	The number of pages visited during the session	Integer numbers	D
True positives/time	The number of true positive reports per unit of time	Real numbers	D
Explored pages/time	The number of pages explored per unit of time	Real numbers	D
Precision/time	The bug report precision per unit of time	Real numbers	D
Recall/time	The bug report recall per unit of time	Real numbers	D

Moreover, thanks to the experiment, we identified one additional bug in the PassAndroid AUT that we were not aware of at the time of bug injection.

After the experiment, each of the bugs reported by students was labelled as True Positive (TP) or False Positive (FP) by a member of our team. The labelling protocol can be expressed as follows:

- A bug is labelled as TP if it was originally present in the list of known bugs or if it is possible to reproduce it within the AUT after the experiment.
- A bug is labelled as FP if it was not originally present in the list of known bugs and it is not possible to reproduce it within the AUT after the experiment.

All the other bug reports (e.g. bug reports for which the description was not clear or bug reports which refer to the testing application rather than the AUT) were discarded.

3.5 Analysis procedure

To answer the proposed RQs, we formulated the hypotheses in Table 6.

We performed the statistical analysis through R (R Core Team, 2025). We implemented non-linear models for repeated measurement data, specifically with the `nlme` package (Pinheiro et al., 2025).

We analysed each dependent variable separately using a linear mixed-effects model of the following form:

$$X = c_0 + c_T \cdot Treatment + c_A \cdot AUT + c_O \cdot Order + c_S \cdot Sequence + \text{Error}(Subject)$$

Table 5 The list of bugs injected in both AUTs

AUT	Bug ID	Description	Category
PA	BROKEN_SELECT_ICON*	Select icon button is broken	REMOVE_LISTENER
PA	BROKEN_FOOTER*	Add footer image button broken	REMOVE_LISTENER
PA	BROKEN_HEADER*	Add header image button broken	REMOVE_LISTENER
PA	BROKEN_LOGO*	Add logo image	REMOVE_LISTENER
PA	BROKEN_SORT_PASS	Sort type in settings not working	MODIFY
PA	BROKEN_SHARE	Share pass not working	REMOVE_LISTENER
PA	BROKEN_CONDENSED	Condensed mode does not change anything	REMOVE_LISTENER
PA	DUPLICATE_PASS	Create pass creates two passes	DUPLICATION
PA	BROKEN_VOUCHER	VOUCHER breaks category change	REMOVE_LISTENER
ON	BROKEN_INFO	Info not working	REMOVE_LISTENER
ON	BROKEN_PIN	Pin note not working	REMOVE_LISTENER
ON	BROKEN_NOTIFICATION	Notification not working	REMOVE_LISTENER
ON	BROKEN_APPTOUR	App tour not working	REMOVE_LISTENER
ON	BROKEN_SORT_NOTE	Sort not working	MODIFY
ON	DUPLICATE_NOTE	Reduced view duplicates notes	REMOVE_LISTENER
ON	MISSING_STATISTICS	Some statistics are missing (only titles)	REMOVE_LISTENER
ON	BROKEN_FILTER	Category filter not working	DUPLICATION
ON	BROKEN_CHECKLIST_OPTIONS	Options related to checklists are not memorized	REMOVE_LISTENER

Table 6 Correlation and p-values for Precision and Recall

Name	Hypothesis
$H_{pr,0}$	Gamification has no statistically significant impact on Bug Report Precision and Recall.
$H_{c,0}$	Gamification has no statistically significant impact on page exploration.
$H_{tpt,0}$	Gamification has no statistically significant impact on the number of TP bug reports per unit of time.
$H_{prt,0}$	Gamification has no statistically significant impact on precision and recall per unit of time.
$H_{ct,0}$	Gamification has no statistically significant impact on the number of explored pages per unit of time.

In this formula, X represents the examined dependent variable, $c_{\cdot}$ represent the regression coefficients estimated by the model for the corresponding independent variables, and *Treatment*, *AUT*, *Order* and *Sequence* are the independent variables. *Subject* is the subject ID variable. The model was fitted independently for each dependent variable; therefore, no aggregation or weighting of different metrics was performed. For RQ3, we adopted a descriptive statistics approach based on a diverging bar chart, which is more appropriate for a questionnaire (GGLikert, n.d.).

Since we produce tests for seven dependent variables, we choose to apply the Bonferroni correction method (Bland & Altman, 1995) , to compensate for the family-wise error rate. Even though this approach is still object of debate, Perneger (1998) , it is recommended in particular when seeking for connections without a preset set of hypotheses backed by a consistent theory, and the tests are not independent (Armstrong, 2014). The Bonferroni correction method consists in dividing the significance level value (α) by the number of hypotheses being tested (in our case, the number of dependent variables). With an initial significance level of 0.05, we obtain a corrected significance level of:

$$\alpha_c = \alpha/7 = 0.05/7 = 0.007$$

4 Results

In this section, we present our results related to effectiveness, efficiency and engagement. After a first data cleaning step, 159 reports were chosen for the analysis. The excluded results are mostly due to issues that occurred during the live experiment, which made it impossible for the student to complete the two sessions and therefore upload the required artefacts.

4.1 RQ1 - Effectiveness

Table 7 shows the coefficients and the p-values obtained while computing the correlation between precision, recall and explored pages against the considered factors.

Looking at the p-values, we cannot refuse $H_{pr,0}$ for treatment (Trtm). All the other factors do not have a statistically significant impact on precision or recall. A remarkable result related to effectiveness is the fact that students discovered a bug in one of the two AUTs (namely PassAndroid) that was already present before the bug injection – however, this

Table 7 Correlation and p-values for Precision, Recall and Explored Pages

	Precision		Recall		Explored pages	
Factor	coeff.	p-value	coeff.	p-value	coeff.	p-value
Seq_2	−0.0026	0.9459	−0.0027	0.8026	3.4043	0.0431
Trtm_g	0.0011	0.9757	−0.0187	0.0799	8.5054	$<$ **0.0001**
AUT_{pa}	0.0950	0.0129	0.0233	0.0305	−3.0152	0.0729
Ord_{GV}	0.0383	0.4501	0.0056	0.6970	−1.4429	0.5680

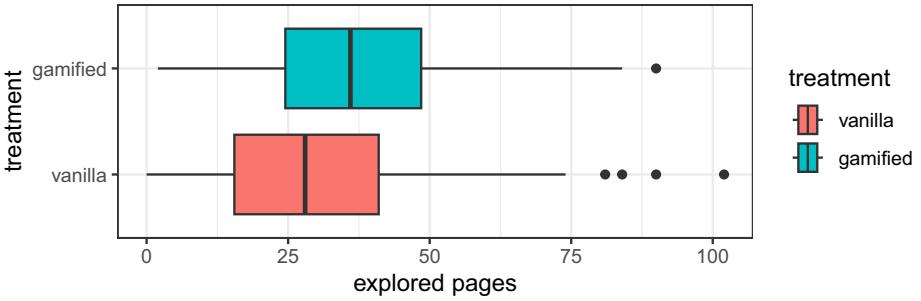

Fig. 4 Page exploration

additional bug has no significant impact on the computed statistics, since it was reported by a rather small number of participants (only 6 students).

We report a significant positive effect on page exploration related to the usage of Gamification ($p =< 0.0001$), which allows us to reject $H_{c,0}$. A noticeable result is the significant positive effect of the sequence session number (2) on the explored pages ($p = 0.0431$). Figure 4 presents a boxplot graph showing the distribution of page exploration results, grouped by the treatment factor.

4.2 RQ2 - Efficiency

Related to the efficiency, we define three metrics: number of TP bug reports per unit of time, Precision and Recall per unit of time, and number of VPs visited per unit of time. In Table 8 we report the coefficients and p-values for the regression analysis performed on TP, Precision, Recall and Explored pages per unit of time. We do not experience significant effects on TP/time, precision/time, nor recall/time. The explored pages per unit of time, instead, is positively influenced significantly by the sequence number (2) and by the application of Gamification. Order and AUT have no significant impact.

Figure 5 shows the effect of the treatment on the page exploration per unit of time.

4.3 RQ3 - Engagement

Regarding the perceived engagement of participants in the experiment, we collected answers from 159 participants. Among the participants, 120 answered all the questions, while 38 missed one or more questions.

The list of questions is reported in Table 9. The questionnaire was the same for both sessions; however, the gamified session included three more questions specific to the gamified

Table 8 TP, precision, recall and explored pages per unit of time

	TP/Time		Precision/Time		Recall/Time		Explored pages/Time	
Factor	coeff.	p-value	coeff.	p-value	coeff.	p-value	coeff.	p-value
Seq_2	0.0004	0.9478	0.0002	0.9191	0.0001	0.8839	0.3167	**0.0002**
Trtm_g	−0.0086	0.1141	0.0000	0.9905	−0.0009	0.1221	0.3709	$<$ **0.0001**
AUT_{pa}	0.0132	0.0166	0.0047	0.0188	0.0011	0.0621	−0.1671	0.0488
Ord_{GV}	0.0034	0.6188	0.0020	0.4501	0.0003	0.6341	−0.1174	0.3250

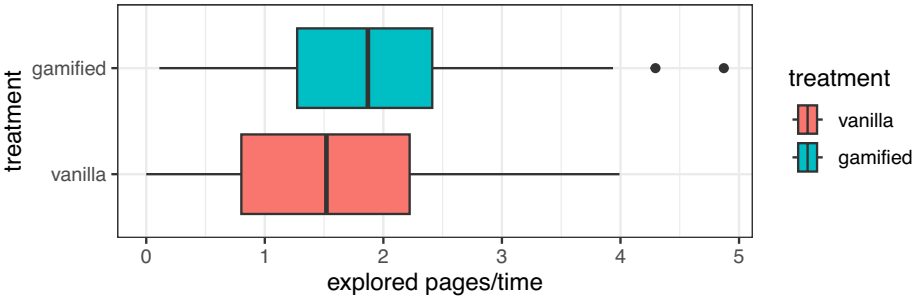

Fig. 5 Explored pages per unit of time

experience (marked with an asterisk (*) in the table). The questions prefixed with "SUS" are related to the usability of the system, while the ones prefixed with "ENG" are related to the engagement.

In Fig. 6, the distribution of the answers to the Vanilla-Likert questionnaire is shown. On the usability side, most of the students reported not feeling like they may use the system for testing activities in future (Q1), and the system is deemed more complex than necessary (Q2). On the satisfaction and engagement side, most of the participants report not having fun while using the system (Q11). Finally, a strong majority (55%) of the students think that gamification mechanics could improve their productivity (Q16).

Figure 7 shows the distribution of the answers to the Gamification-related Likert questions. As can be seen, the number of students reporting their willingness to use the system for testing activities in future rises from 36% to 41% (Q1). In general, the system is found to be more usable than the Vanilla one (Q2 - Q9). Regarding gamification, 48% of the students find the gamification elements clear and understandable (GQ17). A strong majority of them (63%) believe that they would use gamification techniques in future if they had to do Exploratory Testing (GQ18), and that it is a good idea to use gamification for Exploratory Testing (GQ19). It is also interesting to observe that most of the participants (37%) had fun while using the system (Q11). Finally, 58% of the students agree with the fact that gamification mechanics could improve their productivity (Q16), which is in line with the Vanilla-Likert answers.

We also collected answers on the individual gamification features implemented in the testing application. The distribution of the answers is depicted in Fig. 8, while the questions are listed in Table 10 (the questions are the same for each gamification feature). Coverage indicator (i.e. progress bars) was the most appreciated feature, with 83% of the participants agreeing on its usefulness (coverage.Q1) and 76% expressing their appreciation. Instead, customization was deemed the less useful and motivating feature (customization.Q1, customization.Q4).

Table 9 List of questions from the questionnaire

questionID	question
SUS1	I may use this system for testing activities in future
SUS2	The system appeared to be more complex than necessary
SUS3	The system was easy to use
SUS4	I think I would need technical support to use the system
SUS5	The system functionalities were well integrated
SUS6	I think there was too much inconsistency in the system
SUS7	Most of the people could learn to use the system in little time
SUS8	I needed many information before starting using the system
SUS9	The explanations of the instructors were clear enough
SUS10	It was easy to choose which path to explore to test the application
ENG1	I had fun while using the system
ENG2	While using the application I lost the cognition of time
ENG3	I found using the system frustrating
ENG4	I felt autonomous while using the system
ENG5	I felt self-confident while using the system
ENG6	I think that gamification mechanics could improve my productivity
ENG7*	Gamification elements were clear and understandable
ENG8*	If I had to do Exploratory Testing in future I would use gamification techniques
ENG9*	I believe it is a good idea to use Gamification for Exploratory Testing

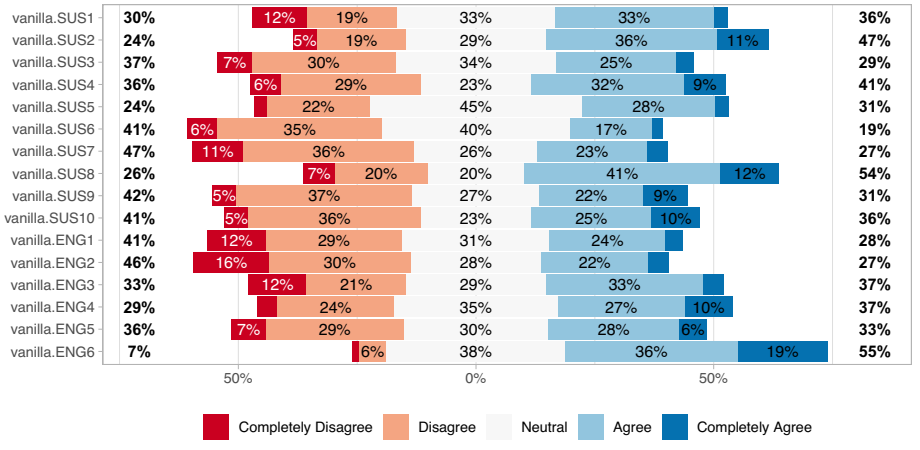

Fig. 6 Likert Vanilla

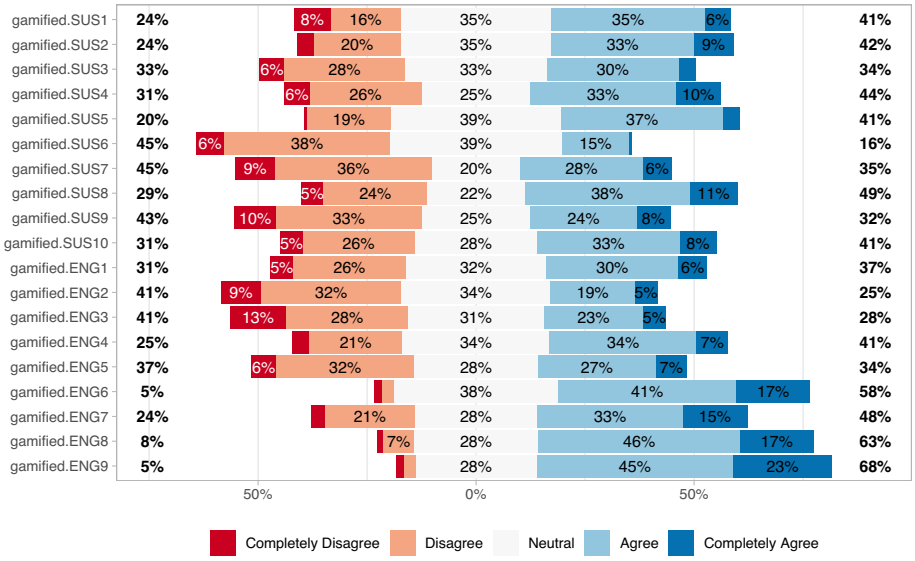

Fig. 7 Likert gamified

To further disentangle usability perceptions from gamification-driven engagement, we computed the correlation between SUS scores and engagement scores across participants, for the gamified experience. The correlation matrix, represented by Table 11, shows that the relationship between usability and engagement items is weak to moderate and non-uniform rather than systematically strong. Across all 90 SUS-ENG pairs, the average absolute correlation is 0.25, with the majority of pairs (62/90) falling below 0.3. A small subset of pairs – notably SUS2/SUS3 with ENG3, ENG4 and ENG5 – show moderate-to-strong correlations ($|r| > 0.4$). We also report the correlation matrix heatmap, for convenience, in Fig. 9. The implications of these results will be discussed in Section 5.

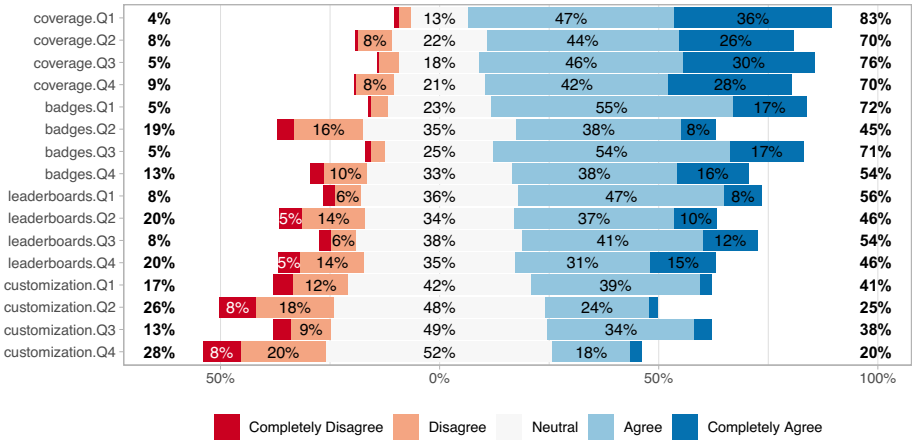

Fig. 8 Likert gamified mechanics

Table 10 Questions related to individual gamification features

ID	Question
Q1	I think that the feature is useful
Q2	I think that the feature influenced my usage of GAppium
Q3	I appreciated the feature
Q4	The feature motivated me doing exploratory testing

Table 11 Correlation matrix among SUS and ENG answers from the gamification questionnaire

	ENG1	ENG2	ENG3	ENG4	ENG5	ENG6	ENG7	ENG8	ENG9
SUS 1	0.51	0.09	−0.40	0.38	0.29	0.23	0.24	0.40	0.33
SUS 2	−0.43	−0.07	0.56	−0.48	−0.43	−0.16	−0.33	−0.32	−0.17
SUS 3	0.38	0.10	−0.58	0.44	0.51	0.14	0.27	0.27	0.26
SUS 4	−0.26	0.05	0.29	−0.35	−0.25	−0.13	−0.32	−0.22	−0.15
SUS 5	0.26	0.13	−0.37	0.32	0.40	0.10	0.26	0.14	0.20
SUS 6	−0.25	−0.14	0.35	−0.30	−0.41	−0.11	−0.23	−0.16	−0.13
SUS 7	0.24	0.18	−0.40	0.28	0.29	0.11	0.22	0.11	0.21
SUS 8	−0.14	0.03	0.19	−0.24	−0.36	−0.07	−0.12	−0.09	−0.12
SUS 9	0.30	0.01	−0.28	0.15	0.32	0.22	0.35	0.17	0.17
SUS 10	0.26	0.02	−0.20	0.34	0.30	0.16	0.30	0.25	0.19

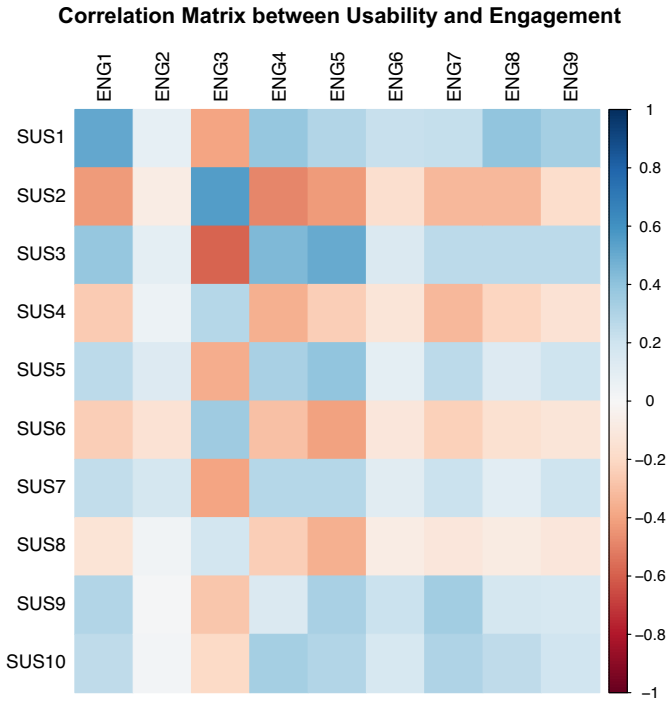

Fig. 9 Heatmap showing the correlation among the SUS and ENG questions

5 Discussion

In this study, we aimed to investigate the impact of gamification on effectiveness, efficiency, and engagement of mobile exploratory testers.

Regarding **effectiveness**, the most notable result is the impact of gamification on exploration. We measured a statistically significant positive correlation between the application of the gamified treatment and the number of pages explored. This suggests to us that gamification has a non-negligible impact on page exploration, therefore serving as a main motivation factor for testers.

We did not measure statistically significant effects of the gamified treatment on the precision and recall. This is in line with our expectations: since bug detection is an activity which requires experience and perspicacity, these skills cannot be acquired thanks to gamification, especially in a short period of time.

Moving to **efficiency**, the most interesting result is the statistically significant positive effect of the gamified treatment on the explored pages per unit of time. We can interpret this result considering gamification as a major driver of motivation for testers. This outcome contrasts with the results of Fulcini and Ardito (2022), who found no significant impact of gamification on efficiency. A possible explanation for this difference might be related to the specific gamification mechanics implemented and how they are presented to the participants. The impact of gamification on True Positive bugs reported per unit of time is not statistically significant. Finally, we witness a non statistically significant effect of gamification on the precision and recall per unit of time. Again, this latter result confirmed our expectations: gamification is not supposed to make testers better at doing their work; it can just serve as a motivational drive.

Engagement was measured with a Likert-scale questionnaire, including generic questions about the system usability and more specific ones related to gamification. Students were asked generic questions taken from System Usability Scale questionnaires, adapted to our use case, and more specific questions about gamification. The most important result is that a strong majority of the students answered that they would use gamification techniques in future if they had to do ET, that it is a good idea to use gamification for ET, and that gamification mechanics could improve their productivity. One major response bias often encountered in surveys is the acquiescence bias (Baxter et al., 2015): participants usually have a tendency to answer positively to questions, without considering the actual content of the question, or their true preference, however the distribution of the other answers suggests that the students were fair enough in their evaluation of the questions – as an example, most of them report experiencing too much inconsistency in the testing tool, both in the vanilla and in the gamified version.

The correlation matrix shown in Table 11 suggests that for some of the considered SUS-ENG pairs (i.e. SUS2/SUS3 with ENG3, ENG4, ENG5) the perceived system consistency and ease of use may be related to specific engagement facets. However, the overall pattern indicates that usability and engagement, as measured, are not interchangeable constructs: most engagement items (ENG2, ENG6, ENG8) show consistently weak correlations with usability across the board. This provides empirical support that engagement, as measured, is not merely a byproduct of perceived usability. Nonetheless, we acknowledge that the addition of gamification elements inherently introduces new interface components, and we cannot fully exclude that part of the observed engagement reflects a reaction to interface novelty rather than the motivational mechanics themselves. We report this as a threat to construct validity in Section 5.1.

5.1 Threats to validity

We report the potential threats to validity of our study according to the four categories described by Wohlin et al. (2012).

Conclusion Validity threats concern issues that can have an impact on drawing the correct conclusion about the relationship between treatments and outcomes. The tests we employed had no statistical prerequisites. Measures were collected automatically; therefore, no human errors should be involved in their collection. Participants were randomly assigned to the groups, and no significant difference exists in terms of expertise. The result may have been impacted by the allocation of participants to the groups; however, given the number of participants, we consider this as very improbable.

Construct Validity threats regard the relationship between theory and observation.

One of the main threats in this sense is represented by the limited time of exposure to gamification mechanics. In fact, other studies have reported the beneficial effect of gamification on knowledge retention (Putz et al., 2020). An hour is too short an amount of time for such an experiment to properly appreciate the effects of gamification on the long-term habits of testers. However, increasing the exposure time is included among our future objectives.

Regarding the adopted metrics, in software engineering, there are no standard metrics to measure effectiveness and efficiency, given their abstract nature. However, the metrics we adopted – explored pages, precision, recall and their derivatives – are widely used in the fields of software engineering and data science (Tan et al., 2018; Washizaki, 2024). A potential concern regarding the use of explored pages as an effectiveness metric is that the increase in the number of pages explored does not necessarily reflect higher-quality testing, as testers could navigate through screens superficially without thoroughly verifying application behavior. In our study, this concern is partially supported by the results: gamification showed a statistically significant effect on the number of explored pages, but no significant effect was observed on bug report precision and recall. This pattern is consistent with the hypotesis that gamification may incentivize broader but not necessarily deeper exploration, possibly as a side effect of mechanics such as progress bars and leaderboards that reward coverage rather than defect detection quality. We acknowledge that the number of explored pages, in isolation, is an imperfect proxy for testing thoroughness, and that the observed coverage gains should be interpreted cautiously. Future studies should consider richer quality indicators, such as fault severity or test case realism (Coppola et al., 2024), and possibly introduce other gamification mechanics that reward defect detection quality as well.

We injected bugs in the two AUTs to measure the precision and recall of the participants, following two principles:

1. The number of injected bugs must be the same in both AUTs.
2. The number of bugs for each class of mutants must be the same in both AUTs.

Where mutant classes are discussed in Section 3. We still have a researcher bias given by the possible difference in discoverability of the bugs present in both AUTs. In fact, the discoverability of bugs is related to their perceived visibility according to the researcher who injected them.

An important limitation of our study relies on our definition of engagement: in fact, engagement is a multidimensional construct comprising behavioral, emotional, and cogni-

tive components (Fredricks et al., 2004). Our measurements primarily capture self-reported perceptions related to emotional engagement, such as perceived enjoyment, interest and motivation, as well as general user experience with the system. Also, we did not adopt a standardized engagement evaluation tool. The relationship with behavioral and cognitive engagement still remain to be investigated in future work, possibly adopting a standardized, validated engagement measurement tool covering all dimensions.

Another potential threat concerns the separation between engagement and usability. Since the gamified tool introduces new visual elements (progress bars, badges, leaderboard, customization options) on top of the baseline interface, part of the variance in perceived engagement could in principle reflect a reaction to interface novelty rather than to the motivational mechanics of gamification per se. To assess the impact of this concern, we further examined the empirical relationship between the two constructs through a correlation analysis (see Section 4), which showed that the majority of SUS-engagement item pairs (69%) exhibit weak correlation ($|r| \leq 0.3$), with an average absolute correlation of 0.25 across all pairs. This suggests that, while not fully orthogonal, engagement and usability were largely perceived as distinct dimensions by participants, supporting the interpretation that the observed engagement effects are not merely a product of interface usability.

We acknowledge, however, that this analysis is correlational and does not allow us to fully rule out interface novelty as a contributing factor to engagement. We assume, consistent with cognitive load theory (Sweller, 1988), that usability acts as a precondition for engagement — i.e., a usable interface frees cognitive resources that can be directed toward the gamified experience — rather than a direct driver of it. This assumption was not formally tested and is reported here as a limitation, to be addressed in future work through designs that isolate interface changes from gamification mechanics (e.g., comparing a gamified version against a non-gamified version with cosmetically equivalent UI additions).

Considering the final questionnaire, there is a potential researcher bias regarding the choice of the questions. Regarding the usability questions, they were all taken from the widely adopted SUS questionnaire. Other questions related to gamification were instead conceived by us. This became necessary to measure the actual impact of gamification on efficiency and effectiveness, and the usefulness and level of satisfaction for each gamification feature.

Internal Validity threats regard potential additional factors that may have an impact on a dependent variable. One potential issue related to the crossover design is the effect of fatigue and learning bias. The experiment was performed in two parallel courses (Italian course, English course). Students from the Italian course performed the experiment in a classroom at Politecnico di Torino in two consecutive sessions, while students from the English course performed the experiment remotely. We did not have control over when remote students performed the two sessions; however, we asked them to perform the two sessions consecutively. We enforced a minimum time requirement of 15 minutes and a maximum time of 30 minutes on the two sessions. The 2x2 factorial design mitigates learning biases by construction. However, since the gamified testing application derives from the vanilla one, the GUI and interaction process are very similar between the two versions. Therefore, we still expect the effect of learning bias on the results of the second testing session. The design of the experiment is supposed to reduce the fatigue effects, which should be experienced during the second session, when participants were evenly split among the two Treatments. Another potential side effect is the carryover effect, which can be experienced when a treatment is

administered right after another, and the effect of the first treatment is still in place. A consequence is the fact that the second treatment may be less effective than expected (Vegas et al., 2016). This effect might be revealed by analysing the impact of order (G-V or V-G) in the analysis. We do not consider the additional point on the final grade as a sufficient incentive for students to potentially have an impact on their motivation level (Jordan, 1986).

Other limitations arise from the testing tool. First, we implemented just a small set of the Octalysis framework's potential gamification mechanics. Other mechanics, like for example online multiplayer features, storytelling, etc., could have been chosen. Second, also considering what has been done in other studies, the level of detail adopted for measuring the exploration level (i.e. pages) may be too high to actually produce a reliable exploration measure. Future improvements of the tool should take this into account by recording also widget-level interactions.

Another limitation is related to the ecological validity of the leaderboard mechanic. Participants competed against simulated rather than real peers, therefore itl remains unclear whether the observed effects can be fully attributed to leaderboard-induced social comparison (Festinger, 1954), as the competitive pressure may not have been perceived as authentic. Our questionnaire does not include items directly assessing perceived competitiveness or leaderboard authenticity. However, participants' responses to the leaderboard-specific items provide partial reassurance: the leaderboard received above-neutral median scores on usefulness (Mdn = 4.0) and appreciation (Mdn = 4.0), and moderate scores on perceived influence on tool usage (Mdn = 3.0) and motivation to perform exploratory testing (Mdn = 3.0), suggesting that the mechanic was at least partially perceived as engaging despite the simulated competition setting. However, we acknowledge this as a residual threat to internal validity: the observed effects may underestimate the true impact of leaderboard-based gamification in settings involving real peer competition. Future studies should include validated items measuring perceived competition (Vorderer et al., 2003), and designs involving real peers to isolate the social comparison mechanism.

External Validity threats concern the generalizability of the results. The selected AUTs are popular open-source Android applications which are currently under development. The applications were selected because of the number of different pages they feature, and the different widgets which they include. A sample of just two applications, however, is too small to guarantee the generalizability of our results. While the selected systems represent real authentic systems, their similarity and limited number reduce the extent to which the findings can be extended to broader mobile development contexts. Variations in application domain, interface organization, architectural design, and overall system complexity may influence how gamification affects testing activities. In particular, characteristics such as the scale of the application, the intricacy of user navigation flows, and the localization of defects could lead to outcomes that differ from those observed in this investigation. We suggest this as future work. Regarding the platform, the testing application is platform-agnostic; therefore, we expect that the same results hold for iPhone apps as well. We injected a fixed set of bugs in both applications. However, these bugs may not be representative of real bugs in mobile applications.

Another major limitation is related to the selection of the sample: it is crucial to acknowledge that the participants were all Master's students, and most of them had limited or no prior hands-on experience in mobile development, software testing or exploratory testing. This novice status acts as a significant moderator on the observed results, since ET is heav-

ily based upon experience. The results derived from this sample are less about reflecting the efficiency of an experienced tester, and more about demonstrating the potential for rapid skill acquisition when exposed to gamification. Moreover, the motivation level of students' can significantly vary from the one of working professionals. These specific issues, related to the sample selection, might constitute a limitation to the generalizability of our results to a non-academic environment.

6 Conclusion

In this paper, we have explored the potential impact of gamification on effectiveness, efficiency and engagement of testers performing mobile exploratory GUI testing. Our main contribution consists of an in-vivo experiment with participants from a master's degree software engineering course. Similar experiments have been run in different contexts (Coppola et al., 2024), however, to the best of our knowledge, this is the first study targeting mobile application exploratory GUI testing, with this degree of freedom. The experiment goal was to extract quantitative and qualitative data to compare the two different treatments applied to participants: vanilla and gamified, in terms of effectiveness, efficiency and engagement. Our study found that subjects exposed to the gamified treatment explored, on average, more pages than those exposed to the vanilla treatment. At the same time, gamification was also found to have a significant positive impact on the explored pages per time unit (a measure of efficiency) of testers. Finally, testers seem to have perceived more emotional engagement with the gamified experience than with the vanilla one. Another result that is worth mentioning is the non-significant impact of gamification on precision and recall, precision and recall per unit of time and true positive bug reports per unit of time. We can interpret these results by saying that gamification can have a significant impact on the short-term motivation of users, but cannot have an effect on tasks which require previous knowledge and experience. Future work should aim at overcoming the limitations described in Section 5. A longitudinal study with a similar experimental design would be beneficial, since it would highlight the long-term effects of the gamified approach. A different sample selection could instead extend the generalizability of the study to other contexts and environments. The testing tool itself could be further improved by adding other gamification mechanics, such as storytelling, multiplayer features, etc. The analysis of engagement could be extended to the other engagement dimensions: behavioral, cognitive and emotional. Finally, a more in-depth behavioral analysis of the testers could reveal hidden patterns in their interaction models, and produce a more robust understanding of the influence of gamification on testers' behavior. The authors are currently working in this direction, to improve the data collection pipeline, and to implement logging of the required data.

Author Contributions L.L. wrote the main manuscript text. All authors reviewed and contributed to the manuscript.

Funding Open access funding provided by Politecnico di Torino within the CRUI-CARE Agreement. This study was carried out within the "EndGame - Improving End-to-End Testing of Web and Mobile Apps through Gamification" project (CUP 2022PCCMLF) – funded by European Union – Next Generation EU Mission 4, Component 1, within the PRIN 2022 program (D.D.104 - 02/02/2022 Ministero dell'Università e della Ricerca). This manuscript reflects only the authors' views and opinions and the Ministry cannot be considered responsible for them.

Data Availability The replication package for the experiment, containing both code and data, is publicly available on Zenodo at the following link: https://doi.org/10.5281/zenodo.20134744.

Declarations

Competing Interests All the authors of the manuscript declare no competing interests.

References

Alegroth, E., Gao, Z., Oliveira, R., & Memon, A. (2015). Conceptualization and evaluation of component-based testing unified with visual gui testing: An empirical study. In *2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)* (pp. 1–10). https://doi.org/10.1109/ICST.2015.7102584

Armstrong, R. A. (2014). When to use the Bonferroni correction. *Ophthalmic and Physiological Optics, 34*(5), 502–508. https://doi.org/10.1111/opo.12131

Bach, J. (2019). *Exploratory Testing*. https://www.satisfice.com/exploratory-testing Accessed 16 Oct 2025

Basili, V. R., & Weiss, D. M. (1984). A Methodology for Collecting Valid Software Engineering Data. *IEEE Transactions on Software Engineering, SE–10*(6), 728–738. https://doi.org/10.1109/TSE.1984.5010301

Baxter, K., Courage, C., & Caine, K. (2015). Chapter 10 - surveys. In K. Baxter, C. Courage, & K. Caine (eds.), *Understanding Your Users (Second Edition)* (pp. 264–301), Second edition edn. Interactive Technologies. Morgan Kaufmann, Boston. https://doi.org/10.1016/B978-0-12-800232-2.00010-9.

Blanco, R., Trinidad, M., Suárez-Cabal, M. J., Calderón, A., Ruiz, M., & Tuya, J. (2023). Can gamification help in software testing education? Findings from an empirical study. *Journal of Systems and Software, 200*, 111647. https://doi.org/10.1016/j.jss.2023.111647

Bland, J. M., & Altman, D. G. (1995). Multiple significance tests: The bonferroni method. *BMJ, 310*, 170. https://doi.org/10.1136/bmj.310.6973.170

Brantley, S., Wilkinson, M., & Feng, J. (2021). Beat the bots: Exploring the effects of placebo manipulation on performance during video gameplay. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 65*(1), 923–927. https://doi.org/10.1177/1071181321651311

Brooke, J. (1996). *SUS - a quick and dirty usability scale* (pp. 189–194)

Bures, M., Frajtak, K., & Ahmed, B. S. (2018). Tapir: Automation support of exploratory testing using model reconstruction of the system under test. *IEEE Transactions on Reliability, 67*(2), 557–580. https://doi.org/10.1109/TR.2018.2799957

Chou, Y. K. (2015). Actionable gamification: Beyond points, badges, and leaderboards. *Createspace Independent Publishing Platform*, South Carolina, USA. https://books.google.it/books?id=jFWQrgEACAAJ

Coppola, R., Ardito, L., Torchiano, M., & Alègroth, E. (2020). Translation from visual to layout-based android test cases: a proof of concept. In *2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)* (pp. 74–83). https://doi.org/10.1109/ICSTW50294.2020.00027

Coppola, R., Fulcini, T., Ardito, L., Torchiano, M., & Alègroth, E. (2024). On effectiveness and efficiency of gamified exploratory GUI testing. *IEEE Transactions on Software Engineering, 50*(2), 322–337. https://doi.org/10.1109/TSE.2023.3348036

Costa, I. E. F., & Oliveira, S. R. B. (2021). A study on assets applied in exploratory test design and execution: An interview application. In *2021 IEEE Frontiers in Education Conference (FIE)* (pp. 1–8). https://doi.org/10.1109/FIE49875.2021.9637359

Deak, A., Stålhane, T., & Sindre, G. (2016). Challenges and strategies for motivating software testing personnel. *Information and software Technology, 73*, 1–15.

Deci, E. L., & Ryan, R. M. (1985). *Intrinsic Motivation and Self-Determination in Human Behavior*. Springer, Boston, MA. https://doi.org/10.1007/978-1-4899-2271-7

Deterding, S., Khaled, R., Nacke, L. E., & Dixon, D. (2011). Gamification: Toward a definition. In *CHI 2011 Gamification Workshop Proceedings* (Vol. 12, pp. 1–79). Vancouver.

Di Martino, S., Fasolino, A. R., Starace, L. L. L., & Tramontana, P. (2021). Comparing the effectiveness of capture and replay against automatic input generation for Android graphical user interface testing. *Software Testing, Verification and Reliability, 31*(3), 1754. https://doi.org/10.1002/stvr.1754

Evers, B., Derakhshanfar, P., Devroey, X., & Zaidman, A. (2020). Commonality-driven unit test generation. In *Search-Based Software Engineering: 12th International Symposium* (pp. 121–136), SSBSE 2020, Bari, Italy, October 7–8, 2020, Proceedings. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-030-59762-7_9 .

Fasolino, A. R., Accetto, C. M., & Tramontana, P. (2024). 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* (pp. 26–29). Gamify 2024. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3678869.3685686 . https://dl.acm.org/doi/10.1145/3678869.3685686 Accessed 2025-02-05

F-Droid - Free and Open Source Android App Repository. https://f-droid.org/ Accessed 28 OCt 2025

federicoiosue: Releases · federicoiosue/Omni-Notes. https://github.com/federicoiosue/Omni-Notes/releases Accessed 05 Nov 2025

Festinger, L. (1954). A theory of social comparison processes. *Human Relations, 7*(2), 117–140. https://doi.org/10.1177/001872675400700202

Fraser, G., Gambi, A., Kreis, M., & Rojas, J. M. (2019). Gamifying a Software Testing Course with Code Defenders. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 571–577). SIGCSE '19. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3287324.3287471

Fredricks, J. A., Blumenfeld, P. C., & Paris, A. H. (2004). School engagement: Potential of the concept, state of the evidence. *Review of Educational Research, 74*(1), 59–109. https://doi.org/10.3102/00346543074001059

Fulcini, T., & Ardito, L. (2022). Gamified exploratory GUI testing of web applications: A preliminary evaluation. In *2022 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)* (pp. 215–222). https://doi.org/10.1109/ICSTW55395.2022.00045. https://ieeexplore.ieee.org/document/9787968 Accessed 13 Jun 2024

Fulcini, T., Coppola, R., Ardito, L., & Torchiano, M. (2023). A review on tools, mechanics, benefits, and challenges of gamified software testing. *ACM Computing Surveys, 55*(14s), 310–131037. https://doi.org/10.1145/3582273

Garaccione, G., Fulcini, T., Stefanut Bodnarescul, P., Coppola, R., & Ardito, L. (2024). Gamified GUI testing with Selenium in the IntelliJ IDE: A Prototype Plugin. In *Proceedings of the 1st ACM/IEEE Workshop on Integrated Development Environments* (pp. 76–80). IDE '24. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3643796.3648459 Accessed 04 May 2026

Gebizli, C.c., & Sözer, H. (2017). Automated refinement of models for model-based testing using exploratory testing. *Software Quality Journal*, *25*(3), 979–1005. https://doi.org/10.1007/s11219-016-9338-2

Ghazi, A. N., Petersen, K., Bjarnason, E., & Runeson, P. (2018). Levels of exploration in exploratory testing: From freestyle to fully scripted. *IEEE Access, 6*, 26416–26423. https://doi.org/10.1109/ACCESS.2018.2834957

Heasman, B., & Gillespie, A. (2019). Participants over-estimate how helpful they are in a two-player game scenario toward an artificial confederate that discloses a diagnosis of autism. *Frontiers in Psychology*, *10* - 2019. https://doi.org/10.3389/fpsyg.2019.01349

Jordan, P. C. (1986). Effects of an extrinsic reward on intrinsic motivation: A field experiment. *The Academy of Management Journal, 29*(2), 405–412.

Laudadio, L., Coppola, R., Torchiano, M., & Fulcini, T. (2025). *GAppium: A Framework to Enact Gamification Mechanics in Appium Inspector*.

Leveau, J., Blanc, X., Réveillère, L., Falleri, J.-R., & Rouvoy, R. (2020). Fostering the diversity of exploratory testing in web applications. In *2020 IEEE 13th International Conference on Software Testing, Validation and Verification* (pp. 164–174). ICST. https://doi.org/10.1109/ICST46399.2020.00026

Leveau, J., Blanc, X., Réveillère, L., Falleri, J.-R., & Rouvoy, R. (2022). Fostering the diversity of exploratory testing in web applications. *Software Testing, Verification and Reliability, 32*(5), 1827. https://doi.org/10.1002/stvr.1827

Ligi (2025) ligi/PassAndroid. https://github.com/ligi/PassAndroid Accessed 05 Nov 2025

Long, Z., Wu, G., Chen, X., Chen, W., & Wei, J. (2020). WebRR: self-replay enhanced robust record/replay for web application testing. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (pp. 1498–1508). ESEC/FSE 2020. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3368089.3417069

Mahmud, T., Che, M., Ngu, A., & Yang, G. (2025). Why android app testing falls short: Empirical insights from open-source projects and a practitioner survey. *Empirical Software Engineering, 30*(6), 163. https://doi.org/10.1007/s10664-025-10726-x

Mårtensson, T., Ståhl, D., Martini, A., & Bosch, J. (2021). Efficient and effective exploratory testing of large-scale software systems. *Journal of Systems and Software, 174*, 110890. https://doi.org/10.1016/j.jss.2020.110890

McCabe, T. J. (1976). A complexity measure. *IEEE Transactions on Software Engineering, SE–2*(4), 308–320. https://doi.org/10.1109/TSE.1976.233837

Mikhaltsov, D., & Sorokin, K. (2022). Detecting anomalous device loads during exploratory testing of mobile applications. In *2022 Ivannikov Ispras Open Conference (ISPRAS)* (pp. 43–49). https://doi.org/10.1109/ISPRAS57371.2022.10076851

Mukherjee, R., Pati, S.K., & Banerjee, A. (2022). Chapter 2 - Performance tuning of Android applications using clustering and optimization heuristics. In S. De, S. Dey, S. Bhattacharyya, & S. Bhatia (eds.), *Advanced Data Mining Tools and Methods for Social Computing. Hybrid Computational Intelligence for Pattern Analysis* (pp. 27–50). Academic Press, Cambridge, Massachusetts, USA. https://doi.org/10.1016/B978-0-32-385708-6.00009-6

Nass, M., Alégroth, E., & Feldt, R. (2019). Augmented Testing: Industry Feedback To Shape a New Testing Technology. In *2019 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)* (pp. 176–183). https://doi.org/10.1109/ICSTW.2019.00048

Negara, S., Esfahani, N., & Buse, R. (2019). Practical android test recording with espresso test recorder. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)* (pp. 193–202). https://doi.org/10.1109/ICSE-SEIP.2019.00029

Neri, G., Marchand, R., & Walkinshaw, N. (2025). Exploratory software testing in scrum: A qualitative study. In S. Peter, M. Kropp, A. Aguiar, C. Anslow, M. I. Lunesu, & A. Pinna (eds.), *Agile Processes in Software Engineering and Extreme Programming* (pp. 160–175). Springer, Cham. https://doi.org/10.1007/978-3-031-94544-1_11

Pedreira, O., García, F., Brisaboa, N., & Piattini, M. (2015). Gamification in software engineering – A systematic mapping. *Information and Software Technology, 57*, 157–168. https://doi.org/10.1016/j.infsof.2014.08.007

Perneger, T. V. (1998). What's wrong with Bonferroni adjustments. *BMJ, 316*(7139), 1236–1238. https://doi.org/10.1136/bmj.316.7139.1236

Pfahl, D., Yin, H., Mäntylä, M. V., & Münch, J. (2014). How is exploratory testing used? A state-of-the-practice survey. In *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering And Measurement. ESEM '14* (pp. 1–10). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2652524.2652531

Pinheiro, J., Bates, D., DebRoy, S., Sarkar, D., Heisterkamp, S., authors, E., Ranke, J., Van Willigen, B., & Team, R. C. (2025). *Nlme: Linear and Nonlinear Mixed Effects Models*. https://cran.r-project.org/web/packages/nlme/index.html Accessed 29 Oct 2025

Plot Likert-type items with 'gglikert()'. https://larmarange.github.io/ggstats/articles/gglikert.html Accessed 29 Oct 2025

Porto, D. D. P., Jesus, G. M. D., Ferrari, F. C., & Fabbri, S. C. P. F. (2021). Initiatives and challenges of using gamification in software engineering: A systematic mapping. *Journal of Systems and Software, 173*, 110870. https://doi.org/10.1016/j.jss.2020.110870

Putz, L.-M., Hofbauer, F., & Treiblmaier, H. (2020). Can gamification help to improve education? Findings from a longitudinal study. *Computers in Human Behavior, 110*, 106392. https://doi.org/10.1016/j.chb.2020.106392

Quality management systems — Requirements. Standard, International Organization for Standardization, Geneva, CH (2015)

R Core Team (2025) R: A language and environment for statistical computing. *R Foundation for Statistical Computing*, Vienna, Austria. R Foundation for Statistical Computing. https://www.R-project.org/

RaiMan's SikuliX (2025). http://sikulix.com/index.html Accessed 17 Oct 2025

Ren, W. (2023). Gamification in Test-Driven Development Practice. In *Proceedings of the 2nd International Workshop on Gamification in Software Development, Verification, And Validation* (pp. 38–46). Gamify 2023. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3617553.3617889

Romdhana, A., Merlo, A., Ceccato, M., & Tonella, P. (2022). Deep reinforcement learning for black-box testing of android apps. *ACM Transactions on Software Engineering and Methodology*, *31*(4). https://doi.org/10.1145/3502868

Shamshiri, S., Rojas, J. M., Galeotti, J. P., Walkinshaw, N., & Fraser, G. (2018). How do automatically generated unit tests influence software maintenance? In *IEEE 11th International Conference on Software Testing, Verification and Validation* (pp. 250–261). ICST. https://doi.org/10.1109/ICST.2018.00033

Stoeve, M., Wirth, M., Farlock, R., Antunovic, A., Müller, V., & Eskofier, B. M. (2022). Eye tracking-based stress classification of athletes in virtual reality. *Proceedings of the ACM on Computer Graphics and Interactive Techniques, 5*(2), 1–17. https://doi.org/10.1145/3530796

Straubinger, P., & Fraser, G. (2023). A survey on what developers think about testing. In *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)* (pp. 80–90). https://doi.org/10.1109/ISSRE59848.2023.00075. https://ieeexplore.ieee.org/abstract/document/10301252 Accessed 14 Nov 2024

Straubinger, P., & Fraser, G. (2024). Engaging developers in exploratory unit testing through gamification. In *Proceedings of the 3rd ACM International Workshop on Gamification in Software Development, Verification, and Validation* (pp. 2–9). Gamify 2024. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3678869.3685683

Straubinger, P., Fulcini, T., Garaccione, G., Ardito, L., & Fraser, G. (2025). Gamifying testing in intellij: A replicability study. *Proceedings of the ACM on Software Engineering*, *2*(ISSTA). https://doi.org/10.1145/3728983

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science, 12*(2), 257–285. https://doi.org/10.1207/s15516709cog1202_4

Tan, P.-N., Steinbach, M., Karpatne, A., & Kumar, V. (2018). *Introduction to Data Mining (2nd Edition)* (2nd ed.). London, England, UK: Pearson.

terryyin: terryyin/lizard. https://github.com/terryyin/lizard Accessed 11 May 2026

Valle, P. H. D., Vilela, R. F., & Hernandes, E. C. M. (2021). Does Gamification Improve the Training of Software Testers? A Preliminary Study from the Industry Perspective. In *Proceedings of the XIX Brazilian Symposium on Software Quality* (pp. 1–10). SBQS '20. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3439961.3440004

Vegas, S., Apa, C., & Juristo, N. (2016). Crossover designs in software engineering experiments: Benefits and perils. *IEEE Transactions on Software Engineering, 42*(2), 120–135. https://doi.org/10.1109/TSE.2015.2467378

Vorderer, P., Hartmann, T., & Klimmt, C. (2003). *Explaining the enjoyment of playing video games: The role of competition*. https://doi.org/10.1145/958720.958735

Wang, Q., Brun, Y., & Orso, A. (2017). Behavioral execution comparison: Are tests representative of field behavior? In *2017 IEEE International Conference on Software Testing, Verification and Validation (ICST)* (pp. 321–332). https://doi.org/10.1109/ICST.2017.36

Washizaki, H. (2024). Guide to the Software Engineering Body of Knowledge (SWEBOK): Version 4.0, 4th edn. *IEEE Computer Society Press*.

Welcome - Appium Inspector. https://appium.github.io/appium-inspector/latest Accessed 28 Oct 2025

Wikipedia (2025) Likert Scale. https://en.wikipedia.org/wiki/Likert_scale

Winkler, A., & Hermann, C. (2019). Placebo- and nocebo-effects in cognitive neuroenhancement: When expectation shapes perception. *Frontiers in Psychiatry*, *10*. https://doi.org/10.3389/fpsyt.2019.00498

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in Software Engineering*. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-29044-2

Authors and Affiliations

Lorenzo Laudadio[1] · Riccardo Coppola[1] · Marco Torchiano[1]

✉ Lorenzo Laudadio
lorenzo.laudadio@polito.it

Riccardo Coppola
riccardo.coppola@polito.it

Marco Torchiano
marco.torchiano@polito.it

[1] DAUIN, Politecnico di Torino, Corso Duca degli Abruzzi, Torino 10125, Torino, Italy