International Journal of Serious Games

ISSN: 2384-8766
https://journal.seriousgamessociety.org/

Article

Evaluation of an industrial case of gamification in software quality improvement

Bilge Say[1], Haluk Altunel[2,3], Mehmet Kosa[4] and Müjde Koca-Atabey[5]

[1]Department of Software Engineering, Atilim University, Ankara, Turkey; [2]Odeon Software Technology, Istanbul, Turkey, [3]Department of Computer Engineering, Bilkent University, Ankara, Turkey; [4]Arizona State University, Department of Psychology, Arizona, USA; [5]Department of Psychology, Ankara Medipol University, Ankara, Turkey
bilge.say@atilim.edu.tr; altunel@bilkent.edu.tr; mkosa@asu.edu; mujde.koca.atabey@gmail.com

Keywords:
Gamification
Software Development
Software Quality
Teamwork
Interpretive Phenomenological Analysis

Received: June 2023
Accepted: July 2023
Published: September 2023
DOI: 10.17083/ijsg.v10i3.594

Abstract

The value of industrial-scale gamification interventions for improving software quality is a topic of interest for software engineering research; but it has not frequently been analysed from the perspective of the developer's experiences. The objective of this study is to qualitatively evaluate developers' experiences in a team-based, leaderboard-style gamification intervention in a large software house. To understand the dynamics of positive outcomes in improving code security and quality, semi-structured interviews were conducted regarding both technical and psychosocial aspects. Eight members of three different leaderboard teams with different standings in the final leaderboard were interviewed, and the transcripts were examined using Interpretive Phenomenological Analysis. The results showed that the gamification intervention did result in positive individual and team-based awareness and behaviour change in a range of technical practices such as unit testing, code reviewing, and design. Post intervention, the participants discussed how their motivation, sense of belonging, and communication improved, also expressing concerns over attainability and fairness of gamification goals and relevance to existing workload. The experiential perspective emerging from analysed themes gives broader insights in technical and socio-psychological dimensions than available in the current literature.

1. Introduction

Lately, within the software engineering sector, there have been many difficulties facing the developers with regards to meeting software quality requirements, especially in terms of implementing a secure software engineering life cycle. All this happens while the recent practices, such as agile and DevOps that emphasise faster and continuous deployment pipelines, are becoming more common in software development for large-scale systems [1]. One way to manage the balance between meeting deadlines and quality expectations is to improve the developers' motivation and engagement. As such, software quality has been one of the target areas of gamification applications in software development. Although recent years have seen an increase in gamification applications in software industry with practitioners as opposed to small-scale studies in software-related university student populations, more studies are needed that empirically evaluate and provide practical results for gamification in industrial-scale software engineering [2]. For this reason, the current study aims to contribute to filling this gap, from a human-factors point of view, by carrying out a qualitative evaluation of participant responses transcribed from semi-structured interviews in a long-term, team- and leaderboard-based gamification intervention for code security and quality improvement in a large software house. Our main goal is to be able to find and analyse important and distinctive technical and social themes in the perceptions and experiences of team members, who took part in the leaderboard-based gamification, by using a specific qualitative analysis method called Interpretative Phenomenological Analysis [3]. To reach the aims mentioned here, the observations concerning the leaderboard experience of the employees are based on the following two main research questions:

RQ1: What are the technical benefits and impediments brought about by the leaderboard for improving software code security and code quality?

RQ2: What are the socio-psychological impacts of the leaderboard on the individuals and their teams with respect to factors such as motivation, engagement, communication, and socialisation?

The rest of this article details this empirical evaluation study, first introducing gamification in software engineering and the details of the particular gamification intervention.

1.1 Background: Approaches to Gamification in Software Engineering and in the Workplace

Gamification is defined as the "use of game design elements in non-game contexts" [4]. With respect to human-resources, gamification techniques in industrial settings are being used for several purposes including, but not limited to, recruitment, onboarding of new hires, training employees, and employee performance enhancement, in addition to many management areas such as finance, governance, logistics, marketing and so on [5] [6]. At its core, the salient reasons for employing gamification techniques are to improve the motivation, engagement, performance and participation of individuals [6][7]. Although it was shown that there are several cases where gamification has been successfully applied in the workplace and resulted in increased engagement and revenue, the diverse nature of the corporate environment requires research to validate the efficacy of gamification in different settings [7]. Gamification has been examined in the software engineering literature for a while now, and using game elements has been posited to have the potential to improve software processes [8]. Adopting gamification techniques in the software engineering domain has an increasing trend, and there have been on-going experiments to apply gamification both to the parts and the entire software development cycle, such as requirements gathering, development, testing, management, and support processes [2], [9]. It has been found that gamification can have an increased motivational drive for practitioners to work on good coding practices [10]. More specifically, giving continuous feedback results in increased practitioner motivation to work on software quality, in addition to enhancing learning and knowledge transfer [11]. Moreover, adopting gamification techniques have also been reported to improve practitioners' test case writing habits [12]. There have been efforts towards the personalization of gamification systems in software development as well [13].

In addition to individual effects of gamification, many studies have focused on the potential collective effects of gamification, mainly for improving teamwork, communication and overall performance. For instance, it was shown that providing feedback on team performance was positively associated with task commitment and performance [14]. Moreover, it was found that team cohesion, i.e. being/feeling as part of a team, positively affects perceived control, concentration, and enjoyment levels [15]. Additionally, it was shown that gamification approaches could work towards creating a positive atmosphere within teams, and improve knowledge sharing [16] [17]. Although pure cooperation-oriented gamification approaches can be employed with success, it was found that inter-team competitions are more likely to foster enjoyment and participation when compared to pure cooperation or pure competition in the crowdsourcing context [18]. Integrating games into traditionally non-game based settings (i.e. workplace) was shown to have positive consequences [19].

Leaderboards have been shown that they are effective both in terms of team building and team reinforcement [13]. They are also usually found to be one of the most utilised elements in gamification [20], [21]. More specifically, leaderboards can be useful for encouraging meetings, social gatherings, informal communications, technical discussions, and monitoring purposes [22]. They are also beneficial for completing training, improving the relationships between team members, supporting newcomers, introducing additional challenges and incentives, and avoiding excessive work that might result in burnout (e.g., reversed leaderboards where being at the bottom would be the goal). When the leaderboard is at a team level, there is a “dependent competition” where, although there is a competition between teams, the team members need to depend on each other to perform well [23].

Using leaderboards in gamified systems has some potential risks as well. If leaderboards are not carefully constructed, they might not result in the intended results, and can even actively undermine the desired outcomes [24]. For instance, designing over-competitive systems may result in intimidation and pressure, which might consequently decrease motivation [24]. Another threat is that people or teams that are at the bottom of a leaderboard might feel that they are incompetent, which would undermine their purpose. As a result of this, they might completely abandon the gamified initiative. Another potential pitfall is the “fairness” problem. Leaderboards, and how points are scored in it for instance, can be susceptible to criticism from the employers in the organisational settings [5]. Therefore, it is imperative to understand the user perspective in an ecologically valid way.

To sum up, researching leaderboards in organisational settings is important to understand their efficacies and limitations. Although there has been a relatively larger body of literature in the education and learning domains, studies focusing on the workplace gamification still need to catch up [7], [25]. More specifically, additional empirical research is needed to be able to evaluate the pros and cons of implementing gamification in organisational settings [7]. Method-wise, qualitative studies are seldom used in software-related gamification studies [26]. Therefore in this study, we conduct semi-structured interviews with employees who have gone through a leaderboard-based gamification intervention, which was deployed to improve software quality and security at a software development company. We aim to understand individual subjective experiences and the efficacy of the gamified application. We analyse the interviews using Interpretative Phenomenological Analysis, a qualitative method, to the best of our knowledge, not previously used in gamification of software engineering processes in industrial settings. Thus we hope to contribute to the momentum of gamification studies in software engineering by a qualitative evaluation of a leaderboard-based gamification intervention on improving the challenges in software quality including software security in an industrial setting.

1.2 Challenges in Improving Software Quality

A successful multi year initiative, called NaPiRE for “Naming the Pain in Requirements Engineering”, summarises the status quo and challenges in requirements engineering -a subdiscipline of software engineering- by industrial and academic surveys [27]. Unfortunately, as far as the literature is concerned, there appears to be no corresponding initiative for software quality. There are noticeable gaps between academic research and industry practices in software quality assurance and improvement as well as lack of empirical user studies, especially concerning heterogeneous, diverse, and real-world software systems [28], which take socio-technical issues into account [1].

A critical feature of improving software quality is improving security. In fact this was one of the main motives behind the gamification intervention in the current study. A recent survey of Finnish agile software practitioners [29] has shown that there is a discrepancy between the level of use and the perceived security impact of software security engineering practices. The possible reasons were depicted as developers’ limited awareness of security engineering practices, and developers finding some of the practices hard to integrate with agile life cycles. Organisational awareness, cross-functional teams, verification tools, environments, and methodologies have been found in a recent multi-method study to be among the moderating factors that need better understanding by software development organisations attempting to materialise security improvement methods [30]. However, the socio-technical factors that have been found to affect the level of secure practice uses may not be straightforward to implement unless increased engagement and motivation to perform such practices are enhanced in individuals and teams, thus making the domain of secure software engineering a good candidate for gamification [2]. Generalizing to software quality including software security, building and maintaining the motivation around teams for software quality is difficult under fast delivery cycles and ‘under pressure’ situations. Especially shorter delivery cycles undermine the quality and security, and accumulate the ‘technical debt’, which needs to be addressed eventually, and sometimes never done. The adoption of continuous delivery has been found in industrial case studies to negatively affect the quality of the source code, increase the number of security vulnerabilities, introduce code smells and duplicated codes, to be able to improve customer satisfaction-focused metrics, such as delivery rates [31]. The challenges exemplified here make software quality improvement a good candidate for gamification interventions, since gamification can contribute positively by creating a motivational dimension for higher quality software. The main inspiration behind the empirical evaluation in this study is the lack of knowledge related to various socio-technical aspects of improving software quality through a gamification intervention. Research questions to be elaborated in the next section are organised under technical impacts (RQ1) and socio psychological impacts (RQ2) based on the factors detailed in the literature cited.

2. Methods and Setting

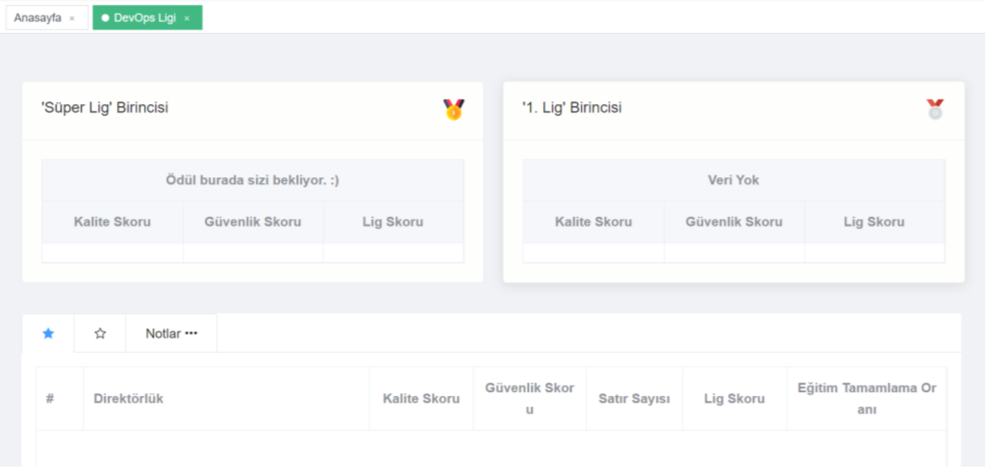
In this section, the industrial setting of the leaderboard-based gamification intervention is introduced, which is followed by the methodological details of the qualitative approach taken in evaluation of the intervention.

2.1 Setting the Context: A Leaderboard-based Gamification Experience for Code Quality Improvement

This study focuses on a large-scale software company that has been developing enterprise-level software, which applied a leaderboard-based gamification system in 2020. The company had 19 product groups as the main departments with a total of 56 product development and operations teams in the banking and finance industry, and a workforce of around 900 people involved in the product development pipeline. The product groups were organised as feature teams; this means that they had front-end and back-end software developers, business analysts, testers, project managers, and product managers. Within each product group, there existed both test automation competencies and DevOps competencies to automate their tests and manage their software packages within the DevOps pipeline. Each product group had 1-6 products as either a big single product or smaller but related products. The company was developing web-based enterprise applications as well as mobile applications in Android and iOS environments. The leading software languages were Java, C#.net, C++, and Python. Even though different languages and technologies were being utilised, the company mainly focused on improving the quality of software and reducing security concerns related to software.

In 2018 and 2019, other attempts were made for quality- and security-related improvements. These trials were based on the yearly performance targets of the product groups. The targets were also added into the company's yearly Balanced Score Card (BSC) [32]. In 2019, 4% of the BSC was reserved for software quality metrics, unit test coverage, and Rule Compliance Index (RCI), and another 4% of the BSC was dedicated for the security measurements, which will be detailed in the upcoming paragraphs. In 2018, a similar approach was utilised with only minor differences in percentages: 5% of the BSC for quality and 3% of the BSC for security. In these two years, the targets were defined as the improvement with respect to the baseline values taken at the beginning of the annual period. For each product group, the baseline values were taken specifically for that product group and the targets were defined as improvements in percentage, which were common for all product groups and company as a whole based on BSC.

In both years, the company reached the targets and had no concern with BSC. When the company examined the outcomes, two patterns emerged. First, if any product group reached its improvement target within the year, that group stopped improving further at that moment and shifted their focus and energy to their other annual targets, especially delivery deadlines. This trend limited the company-wide improvement of software quality and security. Second, product groups were allocating limited time periods for these improvements, mainly at the end of the year. In this way, although software was produced throughout the entire year, improvement was done only at the end of the year. The code base was traced for security and quality assurance, and any issues were fixed. Since such codes were already deployed into production earlier within the year, this backward code improvement operation called for new regression tests. Sometimes the developers resisted changing the code blocks that were running in the production environment. This meant that the software development habits remained unchanged. Therefore, alternative approaches were discussed to gain better software development habits within the company, and leaderboard-based gamification was chosen as the best alternative. The top management supported the idea by approving its application in the company. The leaderboard was organised as a web page, and the data were extracted from the quality and security tools that were integrated. The screenshot of the leaderboard page is shown in Figure 1, where the text and labels are in Turkish as the native language of the developers. In the screenshot, the upper left side panel lists the leader of the league with the quality score, security score and total league score. The upper right panel is listing the leader of the 1st league with the same scores. The bottom panel is listing all the teams with their scores and lines of code. The panels are shown empty for respecting privacy.



'Süper Lig' Birincisi

Ödül burada sizi bekliyor. :)		
Kalite Skoru	Güvenlik Skoru	Lig Skoru

'1. Lig' Birincisi

Veri Yok		
Kalite Skoru	Güvenlik Skoru	Lig Skoru

#	Direktörlük	Kalite Skoru	Güvenlik Skoru	Satır Sayısı	Lig Skoru	Eğitim Tamamlama Oranı

Figure 1. *Leaderboard website (The upper left side panel is listing the leader of the 'super league' with the quality score, security score and total league score. The upper right panel is listing the leader of the 1st league with the same scores. The bottom panel is listing all the teams with their scores and lines of code.)*

The main purpose of the leaderboard-based gamification was to improve the quality and security level of the software produced by the development teams. Additionally, permanent software development behaviour improvements were targeted by increasing the efficiency and effectiveness of the existing software teams in finding security vulnerabilities and developing non-vulnerable code, both individually and as teams. The secondary aim was to develop social aspects such as team belonging and cooperation, both by working and having fun. The development teams were working mainly in the two main offices on the European and Asian sides of Istanbul, and also in Ankara. They started the competition under COVID-19 pandemic restrictions, in which teams were working remotely most of the time.

There were two league seasons within the gamification intervention. The first season was between March and June 2020; the second one was between September and December 2020. In the first season, the main purpose was the elimination of the existing security vulnerabilities within the code blocks. The code base was traced regularly, and the detected security vulnerabilities were prioritised by a security analysis tool as high, medium and low according to their risk level based on the Open Web Application Security Project (OWASP) and reported in the pipeline dashboard [33]. The 19 product groups participated as separate teams in the leaderboard. The product groups were scored in accordance to how many high open vulnerabilities they had in their code base. The lowest number of high vulnerabilities meant leadership in the leaderboard. This motivated the development teams to solve primarily high-vulnerability issues. At the same time, the leaderboards were organised at two levels as in soccer leagues. The main purpose of the two levels is to motivate teams promoting to the super league if they achieve certain goals and keep their motivation to stay there. If you are promoted to the super league your competitors are much stronger. All the product groups started in the 1st league. If a product group could reduce the number of high vulnerabilities to fewer than 10, then they were promoted to the super league, in which the teams would be scored in accordance to how many high, as well as medium, open vulnerabilities they had in their code base. The threshold was set as 10 in line with the company goals since any single-digit value under this threshold is thought to be easier to process and more manageable. This is inline with the numerical cognition literature, which states that single-digit numbers are represented and processed distinctively [34]. At the same time, the company was targeting to decrease the number of high vulnerabilities below 200, which is almost decreasing the number of high vulnerabilities of each product group under 10. Since software development is a continuous task, at the end of each day, the code base was traced with Checkmarx™, which is a static code security analysis tool. The order of the teams was announced to the company on Mondays for three months, after which period the winner product group was announced at the end of June and awarded with a trophy and a coupon certificate for the entire team. At the end of the first season, the overall numbers of high and medium vulnerabilities were reduced by 39.4% and 18.6%, respectively.

In the second season, the same 19 product groups participated in the leaderboard competition as separate teams. The code quality and the code security score were calculated for each product group, accounting for 60% and 40% of the final score, respectively. Code quality was measured as the internal quality of the software based on Sonarqube™, which was utilised for calculating the unit test coverage and the Rule Compliance Index (RCI). The security score was measured again by Checkmarx™ with a difference on the vulnerability density measurement per million lines of code this time. This rule set change was due to the difference in the team size and the Line of Codes (LoC) among the product groups. The second season of the leaderboard was organised at two leagues and similar to the first season. All the product groups started again in the 1st league where, if any product group could reduce the high vulnerabilities to under 10 and increase the unit test coverage to over 40%, it would be promoted to the super league. At this stage, all teams were set in the order of the score defined earlier. Once again, the 40% threshold for unit test coverage was set as it was deemed both achievable and compelling as a target. Once again, the order of the teams was announced to the company on Mondays. Since the focus was on code quality with higher contribution to the leaderboard score, the development teams tried harder to write more unit tests and fix more issues related to RCI. The winning product group was announced at the end of December and was awarded with a trophy. At the end of the second season, the overall unit test coverage and RCI at the company improved by 7.9% and 10.6%, respectively.

2.2 Method

Having two seasons allowed for better security and quality-related improvements than the yearly performance targets set for 2018 and 2019. The leaderboard gamification abandoned the previous patterns followed by teams; i.e., stopping working on quality when the targets were reached, and working on quality only at the end of the year. However, both long-term and implicit technical and sociopsychological impacts of the leaderboard on the individuals were left to be explored and formed the main research focus of this study: What were the qualitative characterizations, such as longevity of the new working patterns, benefits or impediments, of the technical improvements in code security and quality after the whole gamification experience had ended? (RQ1) And how were the developers affected by this experience in terms of socio psychological factors such as their motivation, engagement, sense of belonging, and communication within their teams? (RQ2). Previous studies of gamification in software quality improvement observed increased motivation and engagement to perform quality related tasks, where motivation of developers is normally more geared towards feature development for the software to be developed [2]. However, some developers being frustrated by gamification mechanisms and yet some others resorting to deviant behaviour for increasing their points on leaderboards can also be found in the literature [10]. Some factors, such as team-related perceptions of participants as opposed to individual perceptions are also not well documented. Thus, Interpretative Phenomenological Analysis was chosen as the analysis method, to be able to factor out the experiential factors in the research questions above in a broader sense than the current literature. For that purpose, interviews were organised with the developers who had actively participated in the leaderboard gamification. The product group directors chose the candidate interviewees, hereafter referred to as "participants". The choice of product group directors screening the pool of participants was based on the fact that the directors observed who actively participated in this gamification and contributed actively. The directors, then, asked the most active team members for their volunteership and only the ones who accepted the interviews were included into the study

The participants were eight male employees between the ages of 30 and 38 listed in Table 1. They had various roles within the company as junior and senior software engineers, development team leads, and software architects. The development teams were cross-functional agile teams with different software product responsibilities such as bancassurance, credit card, and digital channels in banking. They had been producing their software in accordance with the Software Product Based Model [35]. Participants were from three different teams: Three from the winning team on the leaderboard, three from the team in second place, and two from the last team on the leaderboard. Before starting the interview process, ethical approval was obtained from the Atilim University's Ethics Committee. The semi-structured interviews were conducted individually via the Zoom platform. All participants were current employees of the software company and the leaderboard was an in-house event, therefore we decided not to record the interviews to be able to provide a more comfortable environment to be able to have more open conversations and gather more quality data. First, third and fourth authors attended the interviews, the fourth author mainly conducted the interviews and first and third authors took notes and asked additional questions where necessary. They ensured full coverage of every detail and established a rich final transcript. The first and fourth authors are female and the second and third authors were male. The authors' specialisations were varied: software engineering (first and second author), game studies (third author) and social psychology, specifically interpretative phenomenological analysis (fourth author). The second author was working in the same company as Director of Strategy and Product Management, so in each interview, he initiated the contact, provided the initial study information but did not attend the interviews. The interviewer provided information about the basic aims of the study, introduced the researchers and any questions from the participants before starting the interviews. The participants provided verbal consent at the beginning of the interviews, which lasted 40 to 65 minutes. (The tentative interview schedule including the semi-structured interview questions are provided in the Appendix). In this way, the leaderboard experiences of the participants were explored in full and comprehensively.

Table 1
The participants and demographics

Pseudonym	Age	Job Title	Seniority	Team
Sinan	38	Software Architect	Senior	Credit Card
Yavuz	33	Senior Software Developer	Mid	Credit Card
Harun	30	Junior Software Developer	Junior	Credit Card
Ayhan	38	Development Team Lead	Senior	Digital Channels
Berk	33	Senior Software Developer	Mid	Digital Channels
Kaan	38	Development Team Lead	Senior	Bancassurance
Tekin	32	Senior Software Developer	Mid	Bancassurance
Mete	34	Senior Software Developer	Mid	Bancassurance

The data was examined using Interpretative Phenomenological Analysis (IPA), which is regarded as a suitable instrument to shed light on individuals' personal and social worlds [3]. This method deals with participants' personally lived experiences and how they make sense of them. The phenomenological element in IPA aims to "give voice" to the participants, while the interpretative one aims to "make sense" of these accounts [36]. It is a double process with the participant and researcher facets, comprising an interpretation process where the participants try to perceive the world and the researcher attempts to comprehend the participants' world. The superordinate and subthemes were decided by the researchers based on the emergent themes that arise through interviews [37]. IPA methodology favours depth over breadth [38]. The emergent themes from interviews were connected in order to reveal the superordinate and subthemes [37]. IPA was originally designed for health psychology research and contributed to the field fruitfully [39]; however, it is also applied to various subject matters including, but not limited to, game addiction [40], problem-based learning in computer engineering education [41], online learning [42], [43] or older adults and technology use [44]. It should be also stated that IPA favours small and homogeneous samples, and special attention is paid to the convergences and divergences of the data [3], [37]. It has also been stated that large sample sizes might be problematic for an IPA study since in-depth analysis is not that possible with larger samples [45]. Therefore an in-depth analysis of a specific experience by eight demographically similar participants was regarded as an appropriate ground for this qualitative methodology. Smith and Osborn [37] stated that doing IPA is inherently flexible, offering relative freedom to researchers in choosing their approaches. Concerning the present research, what we did, as a group of interdisciplinary researchers, was fairly simple; all read the transcripts a number of times and decided on the emergent themes. Then, these emergent themes we discussed and superordinate and subordinate themes were determined, mainly based upon frequency. In the third phase, we matched the most representative transcript excerpts with the themes. Fourth and final, we wrote the results section with the appropriate order of superordinate and subordinate themes and relevant narratives. In addition to the IPA guidelines [3] we also follow the criteria for reporting qualitative research (COREQ) checklist [46], where it fits with the IPA principles. In what follows, we discuss the results.

3. Results

In the IPA, the results are presented without comments, with relevant themes and exemplary comments that best support the theme from the transcripts. The quotes that were presented in this paper were selected based on similarities and differences among the participants. To ease the analysis, the excerpts were listed as separate from the interpretations. The discussion on the themes in their entirety appears in the next section. The interviews and the analysis were conducted in Turkish, and the resulting themes and quotes were translated into English. Any notes by the authors are presented in square brackets, and the participants are referred to by pseudonyms to protect anonymity. The analysis led to the identification of six superordinate themes, with two themes linking further to two subordinate themes. The superordinate and subordinate themes are provided in Figure 2.

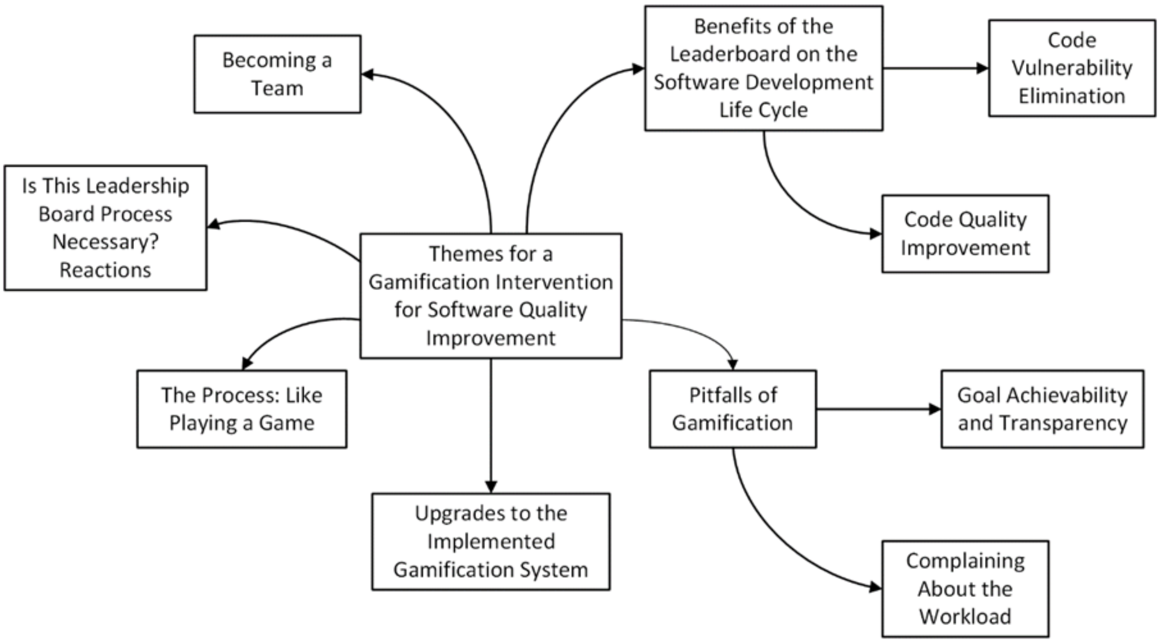

Becoming a Team
Benefits of the Leaderboard on the Software Development Life Cycle
Code Vulnerability Elimination
Is This Leadership Board Process Necessary? Reactions
Code Quality Improvement
Themes for a Gamification Intervention for Software Quality Improvement
The Process: Like Playing a Game
Pitfalls of Gamification
Goal Achievability and Transparency
Upgrades to the Implemented Gamification System
Complaining About the Workload

Figure 2. *Superordinate and subordinate themes map*

The first superordinate theme was entitled by the authors, as “Is this leaderboard process necessary? Reactions”, with the following quotes from the participants:

> We thought that we would not attend this game. We found it funny and unnecessary basically. However, then, we went for it since we thought that it would not be a good idea if we end up at the bottom of the list. Our job is writing codes, being useful…. and there was a reward, so we started immediately (Sinan, 38, Credit Card Team, Software Architect).
>
> We became excited about the process. Competition and reward. These are the issues, which increased the motivation. We worked as a team; teamwork motivated us (Yavuz, 33, Credit Card Team, Senior Software Engineer).
>
> We got the last place, but we were the team that solved the most vulnerabilities. I thought that this process is positive and beneficial. I informed my colleagues. It produced a kind of awareness (Ayhan, 38, Digital Channels Team, Development Team Lead).
>
> These programs have been up and running and in use for many years. In fact, these legacy applications need more care in terms of vulnerabilities since they are programs of 10-15 years of age. Someone did something, and other works were built up on that. (Berk, 33, Digital Channels Team, Senior Software Engineer).

The participants described “The process: Like playing a game”:

> There were slippery issues; it became fun and turned into a competition environment, a game environment, and we socialised. We did so when we included [in the gamification intervention] people who normally socialise by playing World of Warcraft, by going to Kadikoy [a popular district in Istanbul, Turkey] to buy equipment, or by participating in a competition. We wore our athletic uniforms [used as a metaphor]; we actually became like competitive sportsmen (Sinan, 38, Credit Card Team, Software Architect).
>
> The vulnerability sequences were like a game. I was not really focusing on the leaderboard, but it was like playing a game. It was a funny and pleasant experience (Berk, 33, Digital Channels Team, Senior Software Engineer).

> Since there was a reward, it became enjoyable. We did not feel like we were working (Tekin, 32, Bancassurance Team, Senior Software Engineer).

> Within that period, we were having fun; there was a reward, but we were not paying much attention to it. We could not win it, but it is not a big deal (Kaan, 38, Bancassurance Team, Development Team Lead).

The technical benefits were referred to as “The benefits of the leaderboard on the whole Software Development Life Cycle process and code development practices” for instance:

> Our awareness has increased; big issues can be prevented with small measures, and 999 out of 1000 vulnerabilities can be easily resolved (Sinan, 38, Credit Card Team, Software Architect).

> Both using new technologies and doing development and algorithm design studies in parallel were very beneficial in terms of creativity (Harun, 30, Credit Card Team, Junior Software Engineer).

The subtheme of the benefits superordinate theme was “Code quality improvement”:

> The results included unit testing and authentication methods in integration. It brought us the idea of unit testing. Authentication allowed us to figure out which ones are more suitable for us. It enabled us to apply these in parts outside the leaderboard. I think we can write much faster and with higher quality [now] (Yavuz, 33, Credit Card Team, Senior Software Engineer).

> We separated the layers of our project. We properly interpreted the technical faults of each. We set our responsibilities. We found technical bugs and coded each layer to handle the bug (Harun, 30, Credit Card Team, Junior Software Engineer).

> There are many junior developers. There was no code review. Together with the leaderboard, the junior software engineers received their training; their way of writing code has changed (Mete, 34, Bancassurance Team, Senior Software Engineer).

One other subtheme was “Code vulnerability elimination for security”:

> We tried to get rid of the spaghetti code (Yavuz, 33, Credit Card Team, Senior Software Engineer).

> Somewhat, we started from easy to hard with quick gains, retiring parts of the closed code and making a gain in code percentage. We deleted the closed unused code to eliminate vulnerability (Ayhan, 38, Digital Channels Team, Development Team Lead).

> Dividing it step by step and according to the kinds of vulnerabilities, [we found] 30 kinds of vulnerabilities. We separated them according to the type of recurring. We went from the highest to the lowest and remained with only one vulnerability in the end (Mete, 34, Bancassurance Team, Senior Software Engineer).

> We saw that a code used with copy-and-paste is harmful and can be circumvented. We realised that there were code blocks that could be manipulated. We shared it with the team (Berk, 33, Digital Channels Team, Senior Software Engineer).

One other superordinate theme was titled “Becoming a team”:

> Two teams competed fiercely; there was a serious competition in the final few weeks. We did not want to give up now, that we were almost there. After winning, there was a sense of happiness with our friends [team members] (Kaan, 38, Bancassurance Team, Development Team Lead).

> We were like a different company within the company; we did not have much information about the general company aims; working together with other teams had a positive effect on company culture; we gained common goals; we thought we could talk and inquire about things with them. We realised that all teams work together, and that all are visible [in the eyes of the company managers] (Sinan, 38, Credit Card Team, Software Architect).

Feedback was provided as part of possible "Upgrades to the implemented gamification system":

> [In terms of the codebase coverage], prizes should also be given for testing, for example. When incidents happen in tight schedules, with current problem notifications, one can skip the code security or quality issues, I would ensure that this does not happen (Harun, 30, Credit Card Team, Junior Software Engineer).

> I would support the idea that more teams be given some varied [interim] prizes so that some awards are distributed in three-month periods rather than [at the end of] one long marathon; that would encourage more participation. If not, those teams who are in the middle in the leaderboard may not bother [give up half-way] (Ayhan, 38, Digital Channels Team, Development Team Lead).

> The suggestions from the Checkmarx report were 5000 pages - too long. The solution suggestions were not always correct (Mete, 34, Bancassurance Team, Senior Software Engineer).

> The feedback and leaderboard announcement periods [of the intervention organisation team] were sometimes long, and that affected our team negatively (Tekin, 32, Bancassurance Team, Senior Software Engineer).

There are some "Pitfalls of gamification" which appear as two relevant subthemes. "Goal achievability and transparency" was one of the subthemes:

> The goal for unit test coverage was set at percentages that were unattainable (Kaan, 38, Bancassurance Team, Development Team Lead).

> The goal is 40%, but we are not given the reason; and why are we aiming for 80%? I was hearing the why's through the grapevine [company gossip] even though I am a lead (Kaan, 38, Bancassurance Team, Development Team Lead).

The impact on the workload, the other subtheme of "Pitfalls of gamification" is "Complaining about the workload", is shown in the following quotes:

> We had delivery deadlines. We tried to finish the daily workloads. We did not have much time to write secure codes. The effects of the rewards will be short-lived. If I had that authority, I would design teamwork [activities] for [increased] awareness. I would include the managers in the process (Kaan, 38, Bancassurance Team, Development Team Lead).

> I wish that the leaderboard process would continue, but it came with a workload (Mete, 34, Bancassurance Team, Senior Software Engineer).

> The leaderboard did not match with the workload. The plans should be buffered. Those who do 9 units of work should be able to plan so that they can allocate 1 extra unit for the

> team to the league. That should become the company's approach (Tekin, 32, Bancassurance Team, Senior Software Engineer).
>
> It should be a part of the workload. When it was overtime, it became a complete burden and I could not be productive at all. We came up with beneficial outcomes for the customers, and we shared them with the other teams. However, we need to adapt the process [code security processes] to our lives; it should not be a part of an assignment (Berk, 33, Digital Channels Team, Senior Software Engineer).

4. Discussion and Threats to Validity

The themes were grouped and analysed with respect to all transcripts, followed by the specific conditions and limitations pertaining to the study that might have acted as threats to the validity, along with the mitigations made, where possible.

4.1 Analysis of the Themes

As seen in the previous section, six superordinate themes and four subthemes emerged from our analysis. The first superordinate theme was about the necessity in the first place, namely "Is this leaderboard necessary?" This suggests that not everyone might be on board with gamification applications in software companies. Our data showed that some participants might not take the activities seriously or might think that it inhibits regular duties. A study showed that 66% of the population in the U.S. played games in 2018 [47]. Concerning Turkey, the rate for 2021 was 50% according to the Turkish Gaming Market Report [48]. Despite an annual increase in these figures, a considerable part of the Turkish population may not be habitual gamers. Thus, it is possible to claim that there are some software developers who do not engage in playing games and, by extension, may not enjoy such gamified approaches. Moreover, this finding might also stem from daily gamers as well when gamification is applied with a "blanket approach". Games are voluntary activities and they are usually fun when people engage in them voluntarily and not by force. Therefore, coercive participation, regardless of the prior gaming experience of the employee, can be a negative or unnecessary experience [49].

The second superordinate theme was that some participants thought the process was just "like playing a game". Contrary to the first theme, several participants found the application interesting and engaging, as opposed to a routine daily task at work. As expected, the main recurring concept was rivalry. The participants' comments hinted that this type of competition increases motivation, work delivery speed, and sometimes happiness through socialisation and sportsmanship.

The third superordinate theme was found to be about the "benefits of the leaderboard". Some participants stated that they think the leaderboard approach is useful without stating explicit reasons. Some others thought that this approach was useful because they resolved vulnerabilities and code smells in the software much faster, developing strategies on how to best approach such reduction at a reasonable velocity. The gamification intervention also contributed to the renovation projects of the legacy code which were assumed to be functional just because the code was working, and made better use of some additional beneficial tools such as static analysis or authentication tools more frequently. The gamification system helped with the employees' personal development as well. Under this theme, one of the frequent mentions was about the leaderboard intervention significantly increasing awareness throughout the company. In addition to increased awareness around code vulnerability (e.g., unit tests) and quality code (e.g., code review), there were also ad hoc process improvements such as becoming more conscious about design patterns, peer review approvals, and developing portable and generic code writing habits. The benefits of the leaderboard theme can be taken as a superordinate theme to both code quality improvement and code vulnerability elimination. Overall, this theme was in line with other research findings, where leaderboard approaches have been shown to increase engagement, high-quality code, and code readability [50], [51].

"Becoming a team" was the fourth superordinate theme. Although there was ongoing competition on a leaderboard, since the competitors were teams and not individuals, the leaderboard approach brought team members together to work towards a common goal by increasing interaction in a team and boosting collective motivation. The leaderboard approach worked similar to previous research, which showed the efficacy of inter-team competition [18]. This kind of competition had the "unifying" effect where participants felt they shared a common ground in the team and a way to update their knowledge collectively. Additionally, some participants stated that the relationships among the teams were also positively affected. Lastly, the results also suggested that the leaderboard approach can be used to enhance corporate identity.

The participants also recommended some upgrades to the implemented gamification system. One of the main requests was to make it a more holistic approach, including creating a comprehensive game world and lore, a dedicated website, avatars with customization options, and more rewards in numbers (e.g., interim awards) that should be both perceived as achievable and appreciated as a team (e.g., a sponsored dinner with team members and their companions). Another request was to have reports generated by technical tools bearing more beneficial feedback, as well as more instant feedback from the gamified system (e.g., the changes made should be updated in the leaderboard immediately). Such results have been observed for other population samples taking part in gamification-based interventions; for example, a study of the use of gamification-based interventions for dyslexic children had expert evaluation results in line with our specific findings, namely, having a comprehensive game world with varied interim prizes [52]. Although these were some requests from the participants, they still need to be tested empirically and examined how they can be integrated effectively in industrial software development settings.

Lastly, the "pitfalls of gamification" emerged as a superordinate theme, including "goal achievability and transparency" and "complaining about the workload" as subthemes. One of the problems mentioned by the participants was about the achievability of goals. When the employees do not perceive the goals set by the system as achievable, the motivation to participate in the gamified system tends to decline. This is in line with the idea that leaderboards can sometimes create undesired competition, especially among low-performing and/or lowly confident players and teams [53]. This should be remedied by creating achievable goals. Another concern was about the transparency of goals. If the goals are not clear, or if it is not clear why they are set as they are, the motivation towards participation also tends to drop, which our data supports as well. Another common concern was about the gamification approach increasing the workload if the system is not regulated well by the upper management. Gamification implementers in companies should be mindful about the fact that newly introduced gamification approaches may unintendedly force the employees to work more than their regular hours. Therefore, gamification designers should make sure that such activities take place in the regular working hours. Last but not least, the fairness of the system is of utmost importance, since if employees feel that the metrics are designed in such a way that the system induces feelings of injustice, the gamification approach can inhibit the achievement of the desired outcomes rather than facilitate them. In summary, the psychological aspects of the chosen gamification techniques should be carefully examined and implemented [54].

Comparing previous studies of gamification in software engineering, especially in software quality improvement, we can say that our results do corroborate previous findings such as increased playfulness and motivation in participants to achieve quality goals as well as unwillingness of some developers on engaging at the intervention for reasons explained above; some themes, on the other hand, especially at team level such as improvements in team identity and recognition and dealing with the extra workload within the team, and suggestions for organisation of prizes and feedback, had finer details that were not observed in detail or not at all in previous studies [1], [8]-[12],[16], [20], [22].

4.2 Case-specific Circumstances of the Study and Possible Threats to Validity

The leaderboard-style gamification was designed before 2020; it was announced and commenced in early March 2020. Just a few days after the start, on March 11th, the first COVID-19 case was reported within the country [55]. The company shifted to remote working a week after this announcement. Between April and June, the company was working remotely. Besides that, during the same period, there were countrywide restrictions and shutdown measures. The software development teams were working remotely and using collaboration tools, mainly Zoom™. This sudden transition might have had certain effects on the leaderboard experience and the results. However, there was no control group to check for any possible variations among different teams.

Therefore, the entire leaderboard gamification was realised under remote-working conditions. The development teams were responsible for the entire code base of their products. That means, either old or new, all code blocks had to be traced with the Sonarqube™ and Checkmarx™, and the teams were responsible for fixing the vulnerabilities and adding unit tests to all of their code bases. Some code blocks were old, but they had still been up-and-running and categorised as legacy. Within the development teams, the conditions could be such that there were no developers available on some teams who could be considered experienced in terms of handling old code blocks. Therefore, changing or handling a legacy code block in any way could carry the risk of destroying it entirely. Additionally, such legacy code blocks did not exist on an even basis within the product groups; they only appeared randomly in accordance to each product's age.

Another issue was that the size of the teams was not the same within the company, and the product groups included 2-7 development teams and 15-70 individuals in all. Similarly, the code sizes were not the same. Therefore, the difference between the team size and the Line of Codes (LoC) among the product groups might have had some effects on the results.

Next, both seasons of the leaderboard were held during 2020. In that period, the company-wide turnover ratio was 8.17%, which is far lower than the information technology industry average rate of 18.3% [56]. This turnover may have had certain effects on the overall gamification experience. However, there was no control group without any turnover to compare with during that period. The above-mentioned factors, as far as the gamification process itself is concerned, may have affected the empirical design of the intervention, and thus, the phenomenological experiences of the participants. Additionally, our sample happened to be consisting of only males and all were in their thirties, which might have affected the reported experiences. Future studies might consider a more diverse set of participants.

In addition, several issues arise from the evaluation itself. To start with, our participants were not randomly sampled as they had been selected according to their availability at the time of the data collection, and only the ones who consented to contribute were invited. Apart from this, the study took place during the COVID-19 pandemic as explained earlier, which might have also affected some participants' perceptions and responses during the interviews. These two points might additionally limit the generalizability of the findings. Lastly, although each of the four authors separately examined the data and applied the coding procedures, and eventually determined the final themes together, it might still be possible to come up with slightly different interpretations of the data.

5. Conclusions

In this study, we qualitatively analysed the transcripts of eight interviews with participants of a team-based, and leaderboard-style gamification intervention at a large-scale software house for banking and finance, using the IPA methodology. There were two key research questions: The technical effect of intervention on software quality, particularly from a security point of view (RQ1), and the socio psychological impacts on the individuals and their teams (RQ2).

The analysis in our evaluations for RQ1 showed that, as opposed to the problems of having code security and quality improvement in performance cards, gamification processes brought changes in awareness and behaviour, both at the individual and team level. These changes were noted by many of the participants themselves although their teams finished in different positions on the final leaderboard. Writing unit tests, code reviews, cross checks, and using better design patterns were some of such changes. Better-organised teams put emphasis on voluntary participation, taking action by incorporating the investigation of vulnerabilities into their existing agile cycles. Some individuals were annoyed by the conflicts brought about by the extra workload in addition to their delivery schedule; one participant even stated whether the code security and quality culture could be enhanced without resorting to gamification at all.

However, the socio psychological impacts cited (RQ2) were generally positive, with improved intra-team and inter-team communication and motivation. The participants actively sought help from other teams or experts to overcome the vulnerabilities or code quality compliance issues, while enjoying the process in the meantime. The recognition brought about by the leadership board, including the motivation not to finish last, seemed to have given a boost to job engagement and satisfaction, and helped to build a company culture, which might have been harder to maintain and enhance in the COVID-19 period. Game goals were improved for fairness in the second run of the gamification intervention with respect to the size of the code base. The perception of the reachability and fairness of the game goals were evaluated to be important, in addition to timely and digestible feedback in weekly runs. Possible rewards, such as team dinners or technical courses for everyone in the team, as well as more spontaneous game elements during the intervention were suggested by the participants for longer and more active participation of teams.

To the best of our knowledge, IPA has been used for the first time in analysing the experiences of software engineering teams in gamification within an industrial setting. Relatively experiential nature of a small sample of participants (up to 10) is evaluated to be a good starting point for an empirical qualitative evaluation study with a broader focus on various effects on participants as individuals and as team members than available in the literature. Recent studies in gamification in software engineering including those in software quality improvement, mostly concentrate on gamification framework design and less so on experiential evaluation, where evaluation being mostly limited to set of performance outcomes or more common socio psychological expectations from gamification interventions such as positive impact on individual motivation [2], [10], [11], [20]. The authors of this paper have different backgrounds with different levels of experience on qualitative analysis methods and technical aspects of improving software quality. However, such an interdisciplinary mixture positively contributed to the whole process, from its inception and interview preparation to the analysis sessions. Further in-depth qualitative studies on industrial gamification interventions for software quality can complement and extend the current study, such as the use of a multi-phased thematic method (e.g. Grounded Theory) [57].

Acknowledgements

The authors would like to thank all participants for their time and contribution.

Conflicts of interest

The authors have no other relevant financial or non-financial interests to disclose.

References

[1] M. Piattini, I. García-Rodríguez de Guzmán, and R. Pérez-Castillo, “Special issue on quality

management for information systems," *Software Quality Journal,* vol. 28, no. 3, pp. 891-894, 2020. doi:10.1007/s11219-020-09516-z

[2] D. de Paula Porto, G.M. de Jesus, F.C. Ferrari, and S. C. P. F. Fabbri, "Initiatives and challenges of using gamification in software engineering: A systematic mapping", *Journal of Systems and Software*, vol. 173, pp. 1-26, 2021. doi:10.1016/j.jss.2020.110870

[3] J. A. Smith, P. Flowers, and M. Larkin, *Interpretative phenomenological analysis: Theory, method and research,* London: Sage, 2009/2021. https://doi.org/10.1080/14780880903340091

[4] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: defining 'gamification', " in *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments,* September, 2011, pp. 9-15. doi:10.1145/2181037.2181040

[5] R. C., Callan, K. N. Bauer, and R. N. Landers (2015). "How to avoid the dark side of gamification: Ten business scenarios and their unintended consequences," in *Gamification in education and business*, pp. 553-568, Springer, Cham, 2015. doi:10.1007/978-3-319-10208-5_28

[6] V. Wanick and H. Bui, "Gamification in Management: a systematic review and research directions," *International Journal of Serious Games*, vol. 6, no. 2, pp. 57–74, 2019. doi: 10.17083/ijsg.v6i2.282

[7] M. Trinidad, M. Ruiz, and A. Calderón, "A bibliometric analysis of gamification research," *IEEE Access*, vol. 9, 2021, Art. no. 46505-46544. doi:10.1109/ACCESS.2021.3063986

[8] M. Kosa, and M. Yilmaz, (2015). "Designing games for improving the software development process*,"* in *European Conference on Software Process Improvement*, pp. 303-310, Springer, Cham, 2015. doi:10.1007/978-3-319-24647-5_25

[9] O. Pedreira, F. García, N. Brisaboa, and M. Piattini, "Gamification in software engineering–A systematic mapping," *Information and Software Technology,* vol. 57, pp. 157-168, 2015.doi:10.1016/j.infsof.2014.08.007

[10] M. Foucault, X. Blanc, J. R. Falleri, and M. A. Storey, "Fostering good coding practices through individual feedback and gamification: An industrial case study," *Empirical Software Engineering*, vol. 24, no. 6, pp. 3731-3754, 2019. doi:10.1007/s10664-019-09719-4

[11] G. Moser, R. Vallon, M. Bernhart, and T. Grechenig, "Teaching software quality assurance with gamification and continuous feedback techniques," in *2021 IEEE Global Engineering Education Conference (EDUCON),* April, 2021, pp. 505-509. doi:10.1109/EDUCON46332.2021.9453921

[12] P. H. D. Valle, R. F. Vilela, and E. C. M. Hernandes, "Does Gamification Improve the Training of Software Testers? A Preliminary Study from the Industry Perspective," in *19th Brazilian Symposium on Software Quality,* Dec. 2020, pp. 1-10. doi:10.1145/3439961.3440004

[13] M. Muñoz, A. P. Pérez Negrón, J. Mejia, G. P. Gasca-Hurtado, M. C. Gómez-Alvarez, and L. Hernández, "Applying gamification elements to build teams for software development," *IET Software,* vol. 13, no. 2, pp. 99-105, 2019. doi:10.1049/iet-sen.2018.5088

[14] N. Vegt, V. Visch, A. Vermeeren, and H. Ridder, "A case study on gamified interventions for team cohesion in factory work," *Human Technology,* vol. 14, no. 2, pp. 176-208, 2018. doi:10.17011/ht/urn.201808103816

[15] Y. Zhao, M. Srite, S. Kim, and J. Lee, "Effect of team cohesion on flow: An empirical study of team-based gamification for enterprise resource planning systems in online classes," *Decision Sciences Journal of Innovative Education,* vol. 19, no. 3, pp. 173-184, 2021. doi:10.1111/dsji.12240

[16] R. Marques, G. Costa, M. Mira da Silva, D. Gonçalves, and P. Gonçalves, "A gamification solution for improving Scrum adoption," *Empirical Software Engineering,* vol. 25, no. 4, pp. 2583-2629, 2020. doi:10.1007/s10664-020-09816-9

[17] S. Zikos, M. Tsourma, E. E. Lithoxoidou, A. Drosou, D. Ioannidis, and D. Tzovaras, "User Acceptance Evaluation of a Gamified Knowledge Sharing Platform for Use in Industrial Environments," *International Journal of Serious Games*, vol. 6, no. 2, pp. 89–108, 2019, doi: 10.17083/ijsg.v6i2.275

[18] B. Morschheuser, J. Hamari, and A. Maedche, "Cooperation or competition–When do people contribute more? A field experiment on gamification of crowdsourcing," *International Journal of Human-Computer Studies,* vol. 127, pp. 7-24, 2019. doi:10.1016/j.ijhcs.2018.10.001

[19] S. L. Marlow, E. Salas, L. B. Landon, and B. Presnell. "Eliciting teamwork with game attributes: A systematic review and research agenda," *Computers in Human Behavior*, vol. 55, pp. 413-423, 2016. doi:10.1016/j.chb.2015.09.028

[20] C. F. Barreto, and C. França, “Gamification in software engineering: A literature review,” in *2021 IEEE/ACM 13th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, 2021, pp. 105-108. doi:10.1109/chase52884.2021.00020

[21] M. Trinidad, E. Orta, and M. Ruiz, “Gamification in IT service management: A systematic mapping study,” *Applied Sciences*, vol. 11, no. 8, pp. 1-32, 2021. doi:10.3390/app11083384

[22] F. Steffens, S. dos Santos Marczak, C. Treude, L. Singer, D. Redmiles, and B. Al-Ani, “Using gamification as a collaboration motivator for software development teams: A preliminary framework,” in *Anais do XII Simpósio Brasileiro de Sistemas Colaborativos*, Brasil. 2015. Available: https://meriva.pucrs.br/dspace/bitstream/10923/14130/2/Using_Gamification_as_a_Collaboration_Motivator_for_Software_Development_Teams_A_Preliminary_Framework.pdf

[23] N. Vegt, V. Visch, H. de Ridder, and A. Vermeeren, “Designing gamification to guide competitive and cooperative behaviour in teamwork,” in *Gamification in education and business*, Springer, Cham, 2015, pp. 513-533. doi:10.1007/978-3-319-10208-5_26

[24] A. Algashami, L.Vuillier, A. Alrobai, K. Phalp, and R. Ali, “Gamification risks to enterprise teamwork: taxonomy, management strategies and modalities of application,” *Systems*, vol. 7, no. 1, pp. 1-30, 2019. doi:10.3390/systems7010009

[25] A. T. Ferreira, A. M. Araújo, S. Fernandes, and I. C. Miguel, “Gamification in the workplace: A systematic literature review,” in *World conference on information systems and technologies*, Springer, Cham. April, 2017, pp. 283-292. doi:10.1007/978-3-319-56541-5_29

[26] A. Darejeh, and S. S. Salim, “Gamification solutions to enhance software user engagement—a systematic review.,” *International Journal of Human-Computer Interaction*, vol. 32, no. 8, pp. 613-642, 2016. doi:10.1080/10447318.2016.1183330

[27] D. M. Fernández, S. Wagner, M. Kalinowski, M. Felderer, P. Mafra, A. Vetrò, and R. Wieringa, (2017). “Naming the pain in requirements engineering,” *Empirical Software Engineering*, vol. 22, no. 5, pp. 2298-2338, 2017. doi:10.1007/s10664-016-9451-7

[28] A. Orso, and G. Rothermel, “Software testing: a research travelogue (2000–2014),” in *Future of Software Engineering Proceedings*, ACM, New York, NY, USA, 2014, pp. 117-132. doi:10.1145/2593882.2593885

[29] K. Rindell, J. Ruohonen, J. Holvitie, S. Hyrynsalmi, and V. Leppänen, “Security in agile software development: A practitioner survey,” *Information and Software Technology*, vol. 131, pp. 1-13, 2021. doi:10.1016/j.infsof.2020.106488

[30] V. V. Ribeiro, D. S. Cruzes, and G. H. Travassos, “Moderator factors of software security and performance verification,” *Journal of Systems and Software*, vol. 184, pp. 1-22, 2022. doi:10.1016/j.jss.2021.111137

[31] M. Rubert, and K. Farias, “On the effects of continuous delivery on code quality: A case study in industry,” *Computer Standards and Interfaces*, vol. 81, no. 4, pp. 1-20, 2022. doi:10.1016/j.csi.2021.103588

[32] R. S. Kaplan, and D. P. Norton, “The balanced scorecard: measures that drive performance,” *Harvard Business Review*, vol. 83, no. 7, p. 172, 1992. Available: https://hbr.org/1992/01/the-balanced-scorecard-measures-that-drive-performance-2

[33] Open Web Application Security Project. Accessed: July 10, 2021. Available: https://owasp.org/

[34] E. Sixtus, F. Krause, O. Lindemann, and M. H. Fischer, "A sensorimotor perspective on numerical cognition", *Trends in Cognitive Sciences*, vol. 27, no. 4, 2023. doi:10.1016/j.tics.2023.01.002

[35] H. Altunel, and B. Say, “Software product system model: A customer-value oriented, adaptable, devops-based product model,” *SN Computer Science*, vol. 3, no. 1, pp. 1-11, 2022. doi: 10.1007/s42979-021-00899-9

[36] M. Larkin, S. Watts, and E. Clifton, “Giving voice and making sense in interpretative phenomenological analysis,” *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 102-120, 2006. doi:10.1191/1478088706qp062oa

[37] J. A. Smith, and M. Osborn, “Interpretative phenomenological analysis,” in J. A. Smith (Ed.), *Qualitative psychology: A practical guide to research methods*, London: Sage, 2003, pp. 51–80

[38] J. Spiers, and R. Riley, “Analysing one dataset with two qualitative methods: The distress of general practitioners, a thematic and interpretative phenomenological analysis,” *Qualitative Research in Psychology*, vol. 16, no. 2, pp. 276-290, 2019. doi:10.1080/14780887.2018.1543099

[39] R. L. Shaw, “Women's experiential journey toward voluntary childlessness: An interpretative phenomenological analysis,” *Journal of Community and Applied Social Psychology*, vol. 21, no. 2, pp. 151-163, 2011. doi:10.1002/casp.1072

[40] D. Chappell, V. Eatough, M. N. Davies, and M. Griffiths, “EverQuest—It’s just a computer game right? An interpretative phenomenological analysis of online gaming addiction,” *International Journal of Mental Health and Addiction*, vol. 4, no. 3, pp. 205-216, 2006. doi:10.1007/s11469-006-9028-6

[41] C. da Silva Cintra, and R. A. Bittencourt, “Being a PBL teacher in Computer Engineering: An interpretative phenomenological analysis,” in *2015 IEEE Frontiers in Education Conference (FIE)*, Oct. 2015, pp. 1-8. doi:10.1109/FIE.2015.7344234

[42] R. Symeonides, and C. Childs, “The personal experience of online learning: An interpretative phenomenological analysis,” *Computers in Human Behavior*, vol. 51, pp. 539-545, 2015. doi:10.1016/j.chb.2015.05.015

[43] A. Tomovic, “Engagement theory based interpretative phenomenological analysis of computer science students’ perceptions of their learning engagement using mobile devices. (7-2021.)” Doctoral dissertation, St. Cloud State University, 2021. Available: The Repository @ St. Cloud State, https://repository.stcloudstate.edu/

[44] R. Hill, L. R. Betts, and S. E. Gardner, “Older adults’ experiences and perceptions of digital technology: (Dis) empowerment, wellbeing, and inclusion,” *Computers in Human Behavior*, vol. 48, pp. 415-423, 2015. doi:10.1016/j.chb.2015.01.062

[45] J. A. Smith, “Evaluating the contribution of interpretative phenomenological analysis,” *Health Psychology Review*, vol. 5, no. 1, pp. 9-27, 2011. doi:10.1080/17437199.2010.510659

[46] A. Tong, P. Sainsbury, and J. Craig, “Consolidated criteria for reporting qualitative research (COREQ): a 32-item checklist for interviews and focus groups,” *International Journal for Quality in Health Care*, vol. 19, no. 6, pp. 349–357, 2007. doi: 10.1093/intqhc/mzm042

[47] J. Clement, “Number of active video gamers worldwide from 2015 to 2023,” Statista, Accessed: June 1, 2021, Available: https://www.statista.com/statistics/748044/number-video-gamers-world/

[48] Gaming In Turkey, “Turkish Gaming Market Report”, 2021, Available: https://www.gaminginturkey.com/en/turkey-game-market-report-2021-published/

[49] M. Raftopoulos, “Towards gamification transparency: A conceptual framework for the development of responsible gamified enterprise systems,” *Journal of Gaming and Virtual Worlds*, vol. 6, no. 2, pp. 159-178, 2014. doi:10.1386/jgvw.6.2.159_1

[50] R. Kasahara, K. Sakamoto, H. Washizaki, and Y. Fukazawa, “Applying gamification to motivate students to write high-quality code in programming assignments,” in *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, Jul. 2019, pp. 92-98. doi:10.1145/3304221.3319792

[51] Q. Mi, J. Keung, X. Mei, Y. Xiao, and W. K. Chan, “A gamification technique for motivating students to learn code readability in software engineering,” in *2018 International Symposium on Educational Technology (ISET)*, Jul. 2018, pp. 250-254. doi:10.1109/ISET.2018.00062

[52] A. Khaleghi, Z. Aghaei, and M. Behnamghader, “Developing two game-based interventions for dyslexia therapeutic interventions using gamification and serious games approaches entertainment computing journal”, *Entertainment Computing*, vol. 42, 2022. doi:10.1016/j.entcom.2022.100482

[53] F. R. Andrade, R. Mizoguchi, and S. Isotani, “The bright and dark sides of gamification,” in *International conference on intelligent tutoring systems*, Springer, Cham, Jun. 2016, pp. 176-186. doi:10.1007/978-3-319-39583-8_17

[54] R. N. Landers, “Gamification misunderstood: How badly executed and rhetorical gamification obscures its transformative potential,” *Journal of Management Inquiry*, vol. 28, no. 2, pp. 137-140, 2019. doi:10.1177/1056492618790913

[55] COVID-19 pandemic in Turkey. https://en.wikipedia.org/wiki/COVID-19_pandemic_in_Turkey (accessed Sept. 15, 2021).

[56] Tech Trends: Office Needs Evolving for Tech Firms, Available: https://www.facilitiesnet.com/commercialofficefacilities/article/Tech-Trends-Office-Needs-Evolving-for-Tech-Firms--19319 (accessed Sept. 20, 2021)

[57] Z. Masood, R. Hoda, and K. Blincoe, “How agile teams make self-assignment work: A grounded theory study,” *Empirical Software Engineering*, vol. 25, no. 6, 2020, Art. no. 4962-5005. doi:10.1007/s10664-020-09876-x