Investigating Information System Testing Gamification with Time Restrictions on Testers' Performance

Navid Memar
Curtin University,
Bentley, WA - 6102
Australia
navid.memar@postgrad.curtin.edu.au

Aneesh Krishna
Curtin University,
Bentley, WA - 6102
Australia

David A. McMeekin
Curtin University,
Bentley, WA - 6102
Australia

Tele Tan
Curtin University,
Bentley, WA - 6102
Australia

Abstract

This paper presents the results obtained from the evaluation of gamified software testing platform that was developed following series of focus group discussions comprising of software developers and testers. The purpose of this study is to understand the effect of gamification as an additive method that can help improve the performance of software testers. Additionally, in this study, new metrics have been introduced to quantify the performance of software testers fairly and more accurately. Moreover, the effect of time restriction impacting on the performance of software testers will be discussed from results of this study. Findings suggest that the proposed metrics, which more accurately capture the difficulty level of the software code defects, are able to better analyse and compare the performances of software testers in the gamified testing environment. Moreover, results indicated that time restriction may compromise the performance of software testers and the quality of written software test code. On the other hand, results suggest that the performance of software testers in detecting low priority bugs in the gamified software-testing platform was better compared to the other more difficult to detect bugs.

Keywords: gamification; information system; evaluation; fairness

1 Introduction

Information system testing and bug detection practice is still a challenging topic in the information software community. These activities are vital for the successful delivery of software projects. Information system testing helps with assessing the software quality. One of the most crucial goals of software development activities is to achieve a high quality product that meets the estimated budgeted price and be delivered based on the estimated plan (Blum,

1992). Information system testing is a very old concept in the history of digital computers and software products. However, it is expected that this practice will remain in the future as one of the best tools to help with assuring the reliability of software products (Briand, 2007). Additionally, information system testing practice plays an important role in software engineering by consuming 40 to 50% of the entire software development efforts and this rate might be higher for products which require higher level of reliability (Luo, 2001).

With the introduction of fourth generation languages (4GL), there is an acceleration in the implementation process of software products, which requires higher level of maintenance and upgrade of software systems. Thus, for such processes, a proper amount of information system testing activity will be required to ascertain the quality of software products after the changes are made to the product (Marciniak, 1994). With the increased frequency of software releases a higher amount of information system testing will be required to ensure the quality of the software products after changes are made (Luo, 2001). Insufficient testing can lead to various issues such as a need to constantly upgrade applications, economic loss by uncovering software bugs in banking or flight software systems and risking the lives of people due to software failures in vehicles and many other safety related areas (Fraser, 2017). Effective manual or automated testing can be performed to reduce the number of software related risks and disasters. Manual and automated testing approaches can be seen as complementary to each other. Automated testing is capable of performing a large number of tests in a short period of time, whereas manual testing depends on the tester's skills and knowledge for specific cases where automated testing may fail (Leitner, Ciupa, Meyer, & Howard, 2007). Testing requires comprehensive understanding of the program context to provide meaningful test suits while automation can lead to unrealistic tests without clear purpose (Fraser, 2017). To design a useful test suite to detect bugs, an understanding of the intended program behaviour is required. Thus, automation often requires additional human effort to achieve this goal (Fraser, 2017). With manual testing, testers write test suites that may best exercise the program. Whereas automated testing works towards removing the tediousness of the process by the software tool, which generates test cases from the software's specification. On the other hand, information system testing exercise can be a repetitive and mundane activity and often not a very motivating exercise for many software developers (Mäntylä & Smolander, 2016).

Gamification may be a potential solution to these issues by transferring repetitive and boring tasks into fun and motivating activities, providing engaging and competitive environment to perform information system testing practices. This method helps to turn difficult tasks to components of entertaining gameplay, increasing player's motivation and engagement by the use of game elements and mechanics in non-game contexts and using competitive nature of humans to motivate them to compete (Deterding, Dixon, Khaled, & Nacke, 2011; García, Pedreira, Piattini, Cerdeira-Pena, & Penabad, 2017). This method may invoke the element of engagement to the serious activities involved in software development. Thus, repetitive and manual tasks such as maintenance and writing unit test code for the software developers, could become stimulating using rewards and recognition obtained from gamification (de Jesus, Ferrari, de Paula Porto, & Fabbri, 2018). Gamification method has been successfully applied in different domains such as Character Recognition (Von Ahn, Maurer, McMillen, Abraham, & Blum, 2008), Information System Testing (Chen & Kim, 2012), Language Translation (Von Ahn, 2013), Education (Kapp, 2012) and others.

This paper concentrates on the evaluation of the developed gamified software-testing tool to advance software-testing experience with the purpose to provide more engaging and rewarding environment for software testers. Furthermore, this paper presents a new gamified metrics to evaluate the effectiveness of software testers' performances fairly and more accurately. Finally, this paper explores the importance of time pressure on information system testing practice and to evaluate if the proposed tool may improve the software testers' performance.

2 Background

2.1 Gamification and Motivational Factors

Gamification can be defined as the use of game design elements, mechanics, aesthetics, and game thinking in non-game context. The goal of gamification is to improve the level of interest in technology by improving the engagement and motivation of users to perform activities (de Sousa Borges, Durelli, Reis, & Isotani, 2014; Kumar, 2013; Pedreira, García, Brisaboa, & Piattini, 2015; Seaborn & Fels, 2015). Gamification has recently become a buzzword and organizations are often adopting this method to increase their employee's engagement and motivation and to increase the level of intended users' engagement.

To design a gamified system, effectiveness, efficiency, satisfaction, gaming and gamification are the main objectives in the context of gamification (Kumar, 2013). Additionally, collecting, connecting, achievement, feedback, self-expression, reciprocity and blissful productivity are examples of human motivational factors that have applicability to motivate users in real and virtual world (Kumar, 2013). People enjoy collecting recognitions and, in some cases,, they may compare their collections to others, which may enhance the competency of the participants. Some collections may be symbolic such as social statuses and some may have monetary values. Furthermore, people are often motivated when they are part of something larger than themselves. Connecting to different group of people, joining social clubs and socialising are approaches people often take to have meaningful shared experiences and to make life more enjoyable.

On the other hand, people enjoy achievements. Achievement delivers great satisfaction to people. Feedback is another factor that is essential in increasing users' motivation. For instance, software that provides no feedback to their users is not as much enjoyable to use as one that does.

Self-Expression is another important motivation factor for users. Users improve their profiles or players spend time to customize their avatars to control how other users or players view them. Reciprocity is another technique to increase the motivation of users to participate or invest time on specific tasks. For instance, by entering to a store and receiving small free sample, customers might start feeling compelled to purchase the item out of a sense of reciprocity.

Finally, blissful productivity is essential to keep the users motivated to continue progressing. For instance, easy task may cause boredom and when a task is too difficult, it may cause users to be anxious. Therefore, it is important to design tasks appropriately to keep the users motivated. In many studies, players' level of engagement significantly increased after introducing game elements.

There are also other studies that suggest a positive outcome after adopting gamification method. For instance, Barata et al. (Barata, Gama, Jorge, & Gonçalves, 2013) used game elements such as scoring, levels, leader boards, challenges and badges to experiment the outcome of a master's level college course. Authors assessed the impact of gamification on the learning experience by comparing the gamified course and non-gamified version of it, which was practiced the previous year together with the rate of student satisfaction compared to other available courses in the similar academic context. Results suggested a significant increase in lecture attendance, activity and perusing the course reference materials. Furthermore, students indicated that gamification helped in turning complicated materials into interesting, motivating and easier to learn as compared to other courses.

De Freitas and de Freitas (De Freitas & de Freitas, 2013) performed an experiment at the US Air Force Academy by using a software gamification tool. They used Classroom Live technique to streamlines the gamification practice for the trainer by making mutual practices simpler. The tool also provided students with motivating user interface, which provided students with rewards in exchange for their participation. The experience suggested that the gamification tool was successful, and students' respond were positive to gamification.

In (de Jesus et al., 2018), the authors performed a literature review on the use of gamification in the testing context in both education and practice. Authors, retrieved 540 studies in total, however, based on the selection criteria specified by them, 15 studies were finally selected for their literature study. Authors, conducted the literature review for contexts which proposed or applied a technology (approach, tool, framework, method etc.) to gamify information system testing practices and education. The study indicates that the gamification goals were to increase the users awareness regarding their performance and results, boost the adoption of information system testing, motivate the development of creativity to perform testing tasks, minimizing the effort in maintenance, encouraging developers to turn testing practice into a habit, increase the level of software tester's engagement, and many others. Further, they suggested that game elements such as Achievement, Avatar, Badges, Duel, Leader Board, Level, Points, Quest, Social Graph, Team and Virtual good were used in the selected studies to achieve gamification goals.

2.2 Gamification, Serious Game and Game Based Learning (GBL)

Game based learning (GBL) is mainly used in educational domains and it is based on a learning game which has a beginning and an end. Moreover, serious game is another common term used to propose games with the educational intention (Connolly, Boyle, MacArthur, Hainey, & Boyle, 2012). Gamification and GBL are two terms that sometimes could be confusing and cause misunderstanding for many researchers and game designers. (Nah, Telaprolu, Rallapalli, & Venkata, 2013) stated that turning activities or processes into a game by the use of game design elements significant behavioural change can be induced. The intention of Gamification is not to make changes into learning but rather helps with enhancing learning, engagement and positive behaviour by using game design elements (Alsawaier, 2018). Additionally, a serious game is not created with the intention of entertainment rather it is created with a particular intention (Djaouti, Alvarez, & Jessel, 2011). The concept of serious game could be used in different domains such as education, advertisements, politic, etc. In GBL, learners play games in order to learn content but in contrast; gamification involves the use of game elements in an environment outside of digital games. The main difference of

serious games and gamifications could be noticed by knowing that gamified systems are not actual games while serious games are actual games (Djaouti et al., 2011).

2.3 Gamification and Information Testing

The gamification method has been successfully practiced in Information systems and applications in various domains (Hamari, Koivisto, & Sarsa, 2014; Kazhamiakin et al., 2015). Information system testing is often a tedious, monotonous and boring practice (Alrmuny, 2014; Briand, 2007; Shah & Harrold, 2010). Moreover, information system testing is also considered time-consuming and difficult activity to perform (Alrmuny, 2014). However, Gamification can be used as an effective method to help remedy the high level or repetitive tasks during testing. This method may help to increase testing engagement and performance during testing practice. Recently there have been many studies in the field of software engineering by injecting gamification in the serious tasks to raise the level of motivation, performance and engagement of participants in software engineering practice (Pedreira et al., 2015). For instance, (Arai, Sakamoto, Washizaki, & Fukazawa, 2014) introduced gamification as a method to encourage software developers to remove warnings of bug pattern tools and the results indicate that developers were successful in removing 150% warning with the suggested method in compare to the case where participants did not practice the proposed method. Moreover, (Singer & Schneider, 2012) used gamification method to motivate computer science students to deploy additional frequent commits to version control.

In (Johansson & Ivarsson, 2014), the authors explored the effects of gamification on the improvement of unit testing. They used the gamification method to motivate the software testers to perform testing practices. Authors conducted the experiments by including 24 individuals. The effectiveness was evaluated by considering the number of detected bugs as well as requirements covered by the tests. The results suggested a significant improvement for the gamification group in identifying faults while improving level of code coverage by tests written. Further, results indicate that gamification method increased the level of motivation among the testers.

The study conducted by (Arnarsson & Jóhannesson, 2015) used the gamification method to encourage developers to write better unit test codes. Developers' performances were evaluated after analysing both static and dynamic qualities of their written unit test codes. Developers indicated that gamification encouraged them to generate better tests and the tool assisted them in learning about essential information system testing metrics and concepts.

In (Rojas & Fraser, 2016), researchers applied gamification technique to demonstrate mutation testing and to improve testing skills. Code Defenders tool was introduced by researchers to assist practitioners in providing complex mutation testing concepts and to provide a more enjoyable learning experience for students. In Code Defenders players play the role of attackers and defenders to test a java class. The role of attackers is to provide subtle mutants of the java class, which are difficult to detect. The defenders' goal is to create good tests that can detect and isolate the attacker's mutants which can be used as an effective test suite for the unit under test.

2.4 Time Pressure Effects on Task Performance

The effects of time pressure on software development with real (Agrawal & Chari, 2007; Nan & Harter, 2009) and virtual data (Austin, 2001; Valett & McGarry, 1989) at the project level have been discussed in several studies. These research studies often focused on assessing the

project context on large tasks. In (Nan & Harter, 2009), the authors studied the effect of time pressure and their findings suggests a U-shape effect, which is known as Yerkes-Dodson law (Yerkes & Dodson, 1908). The results suggested that, less time pressure can improve the performance of the participants, but with excessive time pressure, the performance drops and has undesirable effects due to increasing level of errors and poor individual commitment to unrealistic expectations. It is also important to identify the effect of time pressure on small tasks. In (Topi, Valacich, & Hoffer, 2005), authors studied the effect of time pressure in the software engineering domain in the small task context on human performance. Their findings on participants' creation database queries indicated that time pressure did not affect task performance. Additionally, another study in the domain of accounting (McDaniel, 1990), evaluated the effect of time pressure on 179 professional staff auditors. The results indicated that time restriction and pressure provided a better efficiency but affected the effectiveness of individual auditors. Furthermore, in (Mäntylä & Itkonen, 2013), the authors studied the effect of crowd size and time restriction in information system testing and results suggested that time pressure on group of five testers using 10 hours in total resulted in detected more defects than a single non time restricted tester using 9.9 hours.

2.5 Relationship Between Performance Evaluation and Job Satisfaction

Many studies have been conducted about the role of social and situational effects on performance-rating process. Performance evaluation plays an important role in human resources systems in organizations. Managers and supervisors' rating of employees' performance leads to serious decisions that are the main influences on a many subsequent human resources actions and outcomes. Indeed, the importance of performance evaluation has resulted in higher efforts to provide a more enhanced performance rating process. The close relationship between job satisfaction and job performance has been studied and addressed previously (Gross & Etzioni, 1985). In (Bartol, Durham, & Poon, 2001) authors explored the effect of rating segmentation on fairness and motivation level. Findings supported that rating segmentation is an important factor as it may influence employees' motivation and work attitude. Additionally, employees often have concerns about their individual performance and how they get evaluated compared to relevant peers (Lyubomirsky & Ross, 1997). The importance of fairness in the organizations and its impact on organisational effectiveness, forms a crucial component and it has been named 'organizational justice research' (Sholihin & Pike, 2009).

Existing organizational literature suggests that justice perceptions regarding the organizational procedures are strongly connected to workplace outcomes such as motivation, commitment of employees and performance (Sholihin & Pike, 2009). It is also suggested that performance evaluation practice is an important factor in organizational justice perception (Sholihin & Pike, 2009). Lau et al. (Lau, Wong, & Eggleton, 2008) studied about the relationship between job satisfaction and fairness of performance evaluation procedures. This study suggested that performance evaluation fairness impacts job satisfaction level by two separate processes, one is the fairness of outcome and the second is from the trust in superior and organisational commitment.

In this paper, we also study existing metrics to evaluate the software testers' performances and argue that how these metrics could be improved by introducing a new gamified metric to evaluate the performance of software testers more accurately.

3 Earlier Work

In this section, a brief discussion on the related works carried out previously is presented. It helps lay the context within which the gamified information system testing platform reported in this study has being developed and evaluated.

3.1 Prototype Implementation

In our initial study (Memar, Krishna, McMeekin, & Tan, 2017), issues such as lack of interest and motivation were identified as barriers for computing graduates (and students) to consider with respect to information system testing. Gamification is the strategy suggested as a potential solution to address these issues (Memar, Krishna, McMeekin, & Tan, 2018). The gamification method has the potential to increase software testers' engagement. Also, it can help remedy the high level of repetition and reduce the boredom level of testers while executing their testing activities.

The serious game principles suggested by Whyte et al. (Whyte, Smyth, & Scherf, 2015) assisted the process of identifying the game elements that are suitable for the gamified information system testing platform. A series of focus group sessions involving software developers and testers assisted in determining the main elements to be applied in the gamified information system testing platform. The main purpose of the study (Memar et al., 2017) was to determine if the design elements and game core elements can help to increase the learning and motivation of software testers.

The core serious game principles are categorised as story line, goal directed learning around targeted skills, feedback and rewards, badges and levels and provision of choice. Story line helps in improving the level of motivation, quality and engagement. Additionally, goal directed learning helps with providing a challenging environment. Furthermore, feedback and awards play a critical part in shaping behaviour in players and serious games. Badges and levels help to provide a challenging, but still achievable level of difficulty for software testers. Finally, provision of choice helps players to be able to have choice in some aspects of the game to have choice over some aspects of the game.

Figure 1 shows the vital game elements to design a gamified information system testing platform. Results suggested that all game core elements are essential for gamifying the information system testing platform. Among the listed categories (Storyline with 33%, Goals Direct Learning Around Targeted Skills with 39%, Increasing Level of Difficulty and Individuation with 33%, Feedback and Rewards Shape Learning with 78% and Provision of Choice with 17%) participants agreed that feedback and rewards are the main factors that help motivate software testers (Memar et al., 2017).

Additionally, Figure 2 presents the essential information required for software testers identified through the focus group sessions. The majority of participants suggested that requirements and design documentation with 86% as well as knowledge with 36% are the key elements required for software testers among the other listed elements (Tools with 29% and Time with 7%) (Memar et al., 2017).

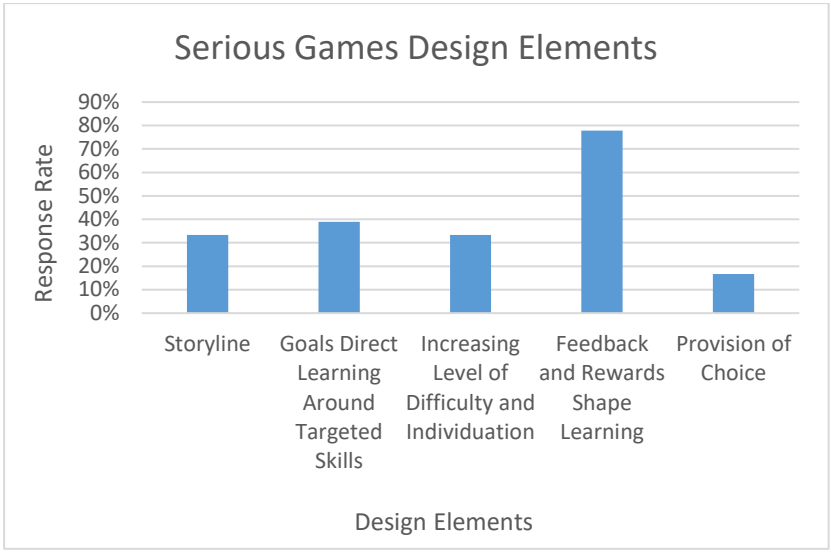

Figure 1. Design elements and motivation (Memar et al., 2017)

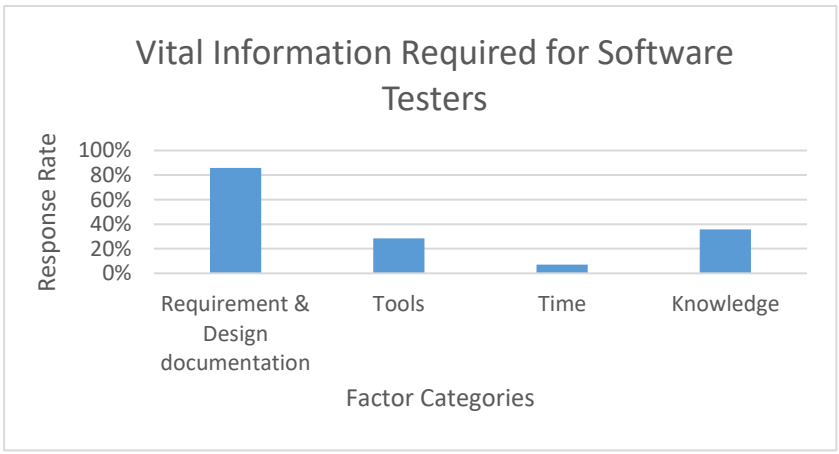

Figure 2. Factors required for software testers (Memar et al., 2017)

3.2 Preliminary Prototype Validation

The developed prototype was subsequently validated by a group of software developers and testers and reported in (Memar et al., 2017). The final product consists of all essential game elements that were identified during the earlier study. Game elements, such as points, real gifts, badges (for difficulty levels), feedback, comparison and provision of choice have been included into the gamified software product in order to increase the level of software testers' motivation and engagement. In the gamified platform, testers require certain amount of points to unlock each badge. Additionally, testers' performance was than evaluated by the review team and appropriate points assigned to each tester. Moreover, the comparison helps to boost the performance of testers after reviewing other testers' performances. For instance, testers will be able to compare their performance against other testers and get motivated to work harder to achieve better results. Furthermore, provision of choice allows each tester to have control over incoming testing requests to either accept or decline the requests. Finally, gifts (rewards) are presented to each tester who succeeded in collecting the appropriate points to unlock the final badge (Memar et al., 2017).

Results of that study indicated that the developed gamified information system testing platform and gamification may be a solution to improve testers' performance, engagement and testing experience. Participants of this study suggested that the developed platform is interesting and motivating for software testers. Furthermore, results suggested that gamification can be used as a tool to increase quality of information system testing activities (Memar et al., 2018).

4 Methodology

As discussed in the background section, there are few studies related to information system testing and gamification. Critical gaps remain in the current framework that need to be addressed to ensure its wider adoption. The objectives of this study are as follow:

1. To identify the perceived effect of time restriction in a gamified information system testing environment on individual software tester performance;
2. Introduce a new metric to evaluate the performance of software testers fairly and more accurately;
3. To identify the effect of gamification on detecting different levels of bugs within the software.

This section explains the chosen method for the evaluation of software testers' performance in a gamified information system testing platform. For this purpose, activities have been discussed in detail.

4.1 Evaluation of Final Gamified Testing Platform

The next step of the research was to first, evaluate the effect of time pressure on the software testers' performance and to identify if time restriction is a factor to motivate the software testers to enhance their productivity. For this purpose, 25 Computing undergraduate students who had experience in software development and software testing as part of their undergraduate course were recruited. Some participants have participated in industry-based software development projects. The Human Research Ethics Committee of Curtin University (approval number HR28/2016) approved the study. Additionally, participants have all been introduced to information system testing technique using the JUnit framework. Participants were briefed on the gamified platform and were given enough time to become familiar with the environment. In total participants were given two tasks. Task 1 consisted of 6 easy, 2 medium and one hard bugs while the second tasks consisted of 2 easy, 4 medium and 2 hard bugs. Participants were briefed about each task and had the opportunity to ask questions during their reading time. Participants were given 15 minutes and 20 minutes to work on task 1 and 2 respectively. Moreover, an additional 5 minute's reading time at the beginning of each task was assigned to help participants be familiar with the tasks to be performed. To ensure that participants were aware of the code contents, comments about the code were added alongside the code. Table 1 presents the number of bugs introduced for each task. Bugs were grouped into three categories: easy, medium and hard.

Tasks	Easy	Medium	Hard
1	6	2	1
2	2	4	2

Table 1. Number of bugs for each task

The following are the description and examples for each bug type:

A) Easy: A bug or mistake in the software that can be fixed by modifying one line of source code, or a bug whose invalid logic resides in one line of source code.

Example 1: `var = "temp"; //` instead of `var = imported_variable`

Example 2: setting the value of a field to the imported value; without validation (i.e. Checking for NULL).

B) Medium: Logical errors that will not function correctly on all occasions.

Example 1: `fuel += amount; //` This refuels the fuel tank by the given amount, but it does not check for invalid fuel amounts (i.e. negative).

Example 2: Not checking array bounds. The logic will work fine until the array is full, and then the program will crash.

C) Hard: Bugs are classified hard if they are both hard to detect, and only trigger on rare cases.

Example 1: In a class named 'Book', writing a page works fine unless the page is full and the number of pages in the book is full.

Example 2: Driving the car works fine in most cases, except that the car should not drive if there is no fuel.

Written questionnaires and recorded discussions were used to establish the evaluation results. In the following section, the findings from this study are explained in detail.

5 Results and Analysis

This section presents the results and findings obtained from the prototype. In the first section of results, the focus is on the qualitative results obtained from the questionnaires during the evaluation session. The given questions are designed to identify the effect of time restriction on software testers' performances. In the second part, the available metrics are defined and new metrics to measure the performance of software testers are introduced. Furthermore, a validation of the proposed metrics together with a comparison of existing and the proposed metrics will be explained. Finally, the last section of results, presents the performance evaluation results of software testers for each level of task difficulty. This comparison helps to identify the effectiveness of gamification based on the level of task difficulty.

5.1 Qualitative Results

In this evaluation, 25 participants, attended the evaluation session and out of them 52% have both Software Development and Software Testing background, 44% have Software Development background and rest have information system testing knowledge. All participants have experience in unit testing technique using the JUnit framework. During the evaluation, students were given an introduction about the gamified platform and had the chance to test the platform and experience the gamified environment. Participants were given the chance to write test codes for 2 different tasks. As discussed earlier, students were given 5 minutes of reading time for each task. Additionally, testers were given 15 and 20 minutes to work on task 1 and task 2 respectively. Tasks were designed in a way for testers to be able to understand the code easily and to be able to write test codes for the given tasks in the given time frame. At the end of the evaluation session, participants were given the questionnaire to

provide feedback on the effect of time restriction on their testing experience. In the questionnaire, participants were given the following question to answer: "What is the effect of time restriction on the information system testing performance in the current gamified information system testing platform?". Results suggested that majority of participants agreed to the fact that time pressure may compromise the performance of software testers. Some of the responses are listed below as ready reference:

- "Having a time restriction may place pressure on the tester, causing the tester to rush."
- "The time constraints make me feel under pressure and I do not think logically due to the stress."
- "The time restriction puts a level of pressure, making you feel rushed, leading to the lack of time to properly read the code and understand what is going on and leading to rushed testing."
- "Time restriction in my opinion is a limiting factor when it comes to software testing. A good software tester is able to get work done in a good amount of time without having an artificial restriction imposed on them. Longer timeframes may improve performance because the tester will be able to write all test cases anyways, but overall, I feel it is diminishing when the timeframe is too low."

Another question was asked on what is the effect of time restriction on the software testing quality in the current gamified information system testing platform and results suggest that everyone agreed that time restriction can compromise the software testing quality.

Some of the responses are listed below as ready reference:

- "Depending on how long the restriction is, just like writing code, with writing test harnesses mistakes can be made if in a rush. And without the proper time to go through it all, the mistakes can get through and create incorrect results."
- "Under time restriction, a subject would be more focused on finding more solutions to a problem rather than elaborating on them. Limited time may leave a subject spending less time on an individual task to ensure they get more tasks done."
- "When having a time restriction it led to feeling rushed, meaning I didn't have time to properly test to a high enough quality as I had to keep in mind of the time. Meaning I could not cover all major test cases, meaning more bugs could slip through."
- "When there is a time restriction, there is pressure to rush and compromise quality for quantity. This leaves room to miss several edge cases and therefore not catch as many bugs."

Furthermore, participants were given the following question to answer: "How likely is time pressure making testers more productive?". The responses suggest that 36% agreed of the fact that time pressure may be a method to increase the productivity of software testers while 32% did not agree based on the fact and the rest of participants were not sure if that would make testers more productive. At the end of the questionnaire, participants were given a chance to provide additional comments or suggestions which some of the responses are listed below as ready reference:

- "If time restriction is ever to have any positive impact, a large amount of research should be dedicated to finding how much time testers should be given for a certain task. Giving too much time or too little time would decrease the performance of the tester, but if a certain limit is reached, it may improve performance. But this limit would be different per tester and per task, so it seems unlikely to be feasible."
- "It really all depends on who the testers are, as some people become more productive with time restraints and others will get stressed out and end up doing less work. Personally, having a flexible time pressure helps me ensure that I try to finish it for a dead line, however if needed can extend the time to get more work done."

5.2 Quantitative Results

5.2.1 Metrics Definition

In the information system testing context, one of the important factors for software testers' performances is to be evaluated fairly and accurately and this may result in higher job satisfaction rate in organisations (a detailed discussion has been provided in the background section). In this section new metrics will be introduced for this purpose. By using the proposed metrics, software testers' performance could be evaluated based on the importance of identified bugs. Additionally, a comparison of new metrics with existing metrics will be provided to validate the proposed metrics. A detailed discussion will be provided to explain why these metrics should be adopted for information system testing performance evaluation to provide more accurate evaluation results. Finally, a detailed discussion about the accuracy and fairness of proposed metrics will be discussed.

1. Effectiveness metric can be calculated with the following formula : E = vdf / vdt while the proposed gamified metric evaluates the performance of testers with the use of following formula: Gamified E= (((0.2*e)/E)+((0.3*m)/M) + ((0.5*h)/H)). In the effectiveness metrics, E is effectiveness, vdf is the number of unique valid defects found, and vdt represents the total number of unique valid defects (Mäntylä & Itkonen, 2013). In contrast, in the gamified effectiveness metric, Gamified E is the effectiveness, e represents the number of unique valid easy defects found, E is the total number of Easy defects, m is the number of unique valid medium defects found, M is the total number of Medium defects, h in the number of unique valid hard defects identified by software testers and H is the total number of Hard bugs existing in the software. This new metric acknowledges the performance of software testers based on the level of detected bugs. For instance, each identified easy bug consists of 0.2 points, while each valid identified medium and hard bug consist of 0.3 and 0.5 points respectively. The new metric provides weight for each particular bug found by the tester depending on the importance of each bug detected. Figure 3 presents the comparison of effectiveness, known as Recall in (Baeza-Yates & Ribeiro, 2011)) and gamified metrics. Results suggest that the correlation coefficient between the two metrics is 0.96656 which shows a strong relation between the two metrics.
2. Although each tester may identify a certain share of unique defects, they additionally may produce a set of invalid bugs (also referred as false positive (Dunsmore, Roper, & Wood, 2003)). The share of valid unique findings among all

findings is called validity (Hartson, Andre, & Williges, 2001) and in the domain of information retrieval, this is commonly referred to as precision (Baeza-Yates & Ribeiro, 2011) and this number is often not reported in in empirical software engineering (Mäntylä & Itkonen, 2013). Validity or precision can be calculated with the following formula: V= (tp / (tp + fp)) where TP is the true positive, (when tester detects a bug and bug exists) and fp represents the false positive (invalid bug reports). In contrast, the proposed gamified validity metric evaluates the performance of testers with the use of following formula: Gamified V = (((0.2*e)/(E + fp))+((0.3*m)/(M + fp)) + ((0.5*h)/(H + fp))). Figure 4 presents the comparison of validity and gamified validity (known as Precision (Baeza-Yates & Ribeiro, 2011)) metrics. Results indicate that the two metrics have a high correlation coefficient with a value of 0.96364. This number supports the fact that there is a strong relation between the two metrics.

3. Lastly, for decision making purposes, the combination measures of effectiveness and validity will be beneficial to evaluate the overall performance of testers more accurately. The importance of this combination measure is in determining better performance results. In the information retrieval domain, both effectiveness and validity are combined in a measure called F-score (Van Rijsbergen, 1986). F score can be calculated with the following formula: Fs = 2 * ((Validity * Effectiveness) / (Validity + Effectiveness)) (Mäntylä & Itkonen, 2013) while, the proposed gamified f score can be calculated as following: Gamified Fs= 2 * ((Gamified Validity * Gamified Effectiveness) / (Gamified Validity + Gamified Effectiveness)). Figure 5 presents the comparison of f score and gamified f score metrics. Results indicate that the two metrics have a strong correlation coefficient with a value of 0.96534.

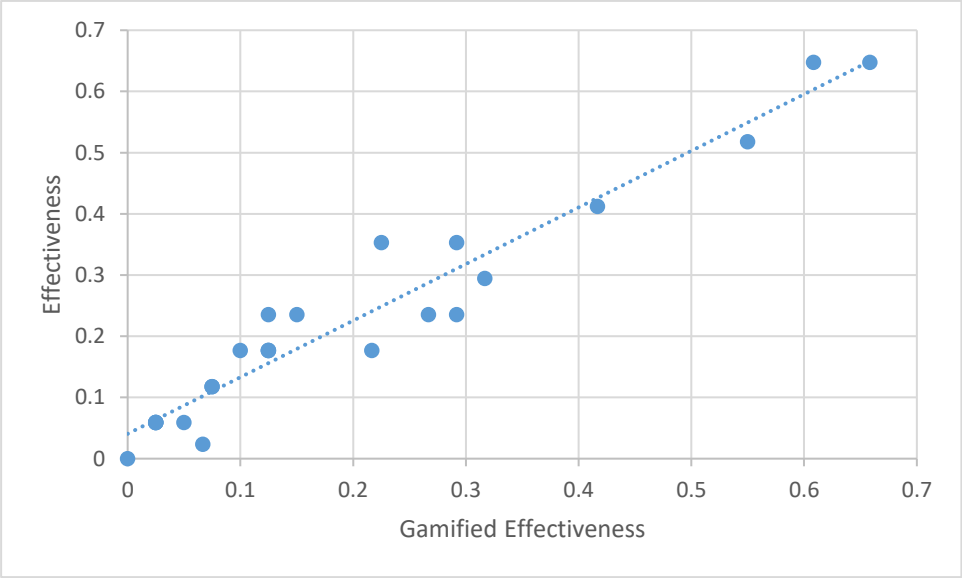

Figure 3. Effectiveness Metric vs. Gamified Effectiveness

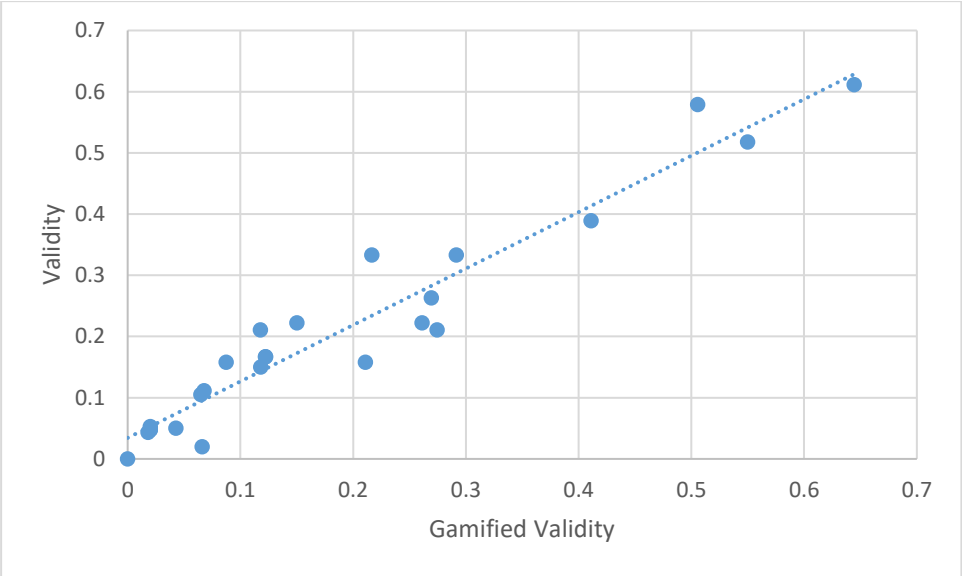

Figure 4. Validity metric vs. Gamified-Validity metric

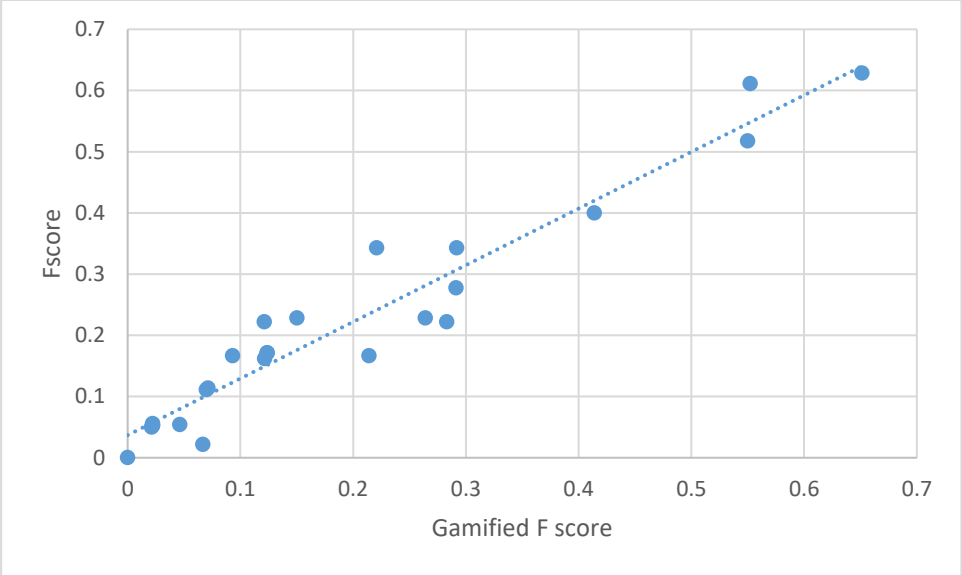

Figure 5. F Score metric vs. Gamified F Score metric

In order to categorise the performance of software testers, gamified effectiveness metric has been selected. The performance of the participants was grouped into three categories of low (performance rate less than 0.2), medium (performance score between 0.2 and 0.5) and high (performance rate greater than 0.5).

Table 2 presents the performance evaluation of participants using effectiveness metric while Table 3 presents the performance of testers using the gamified effectiveness metric. For this purpose, performance of all 25 participants have been evaluated. Findings suggest that Effectiveness metric provide similar performance result for participants that identified the same number of bugs by disregarding the importance of bugs identified by the participants while the proposed gamified effectiveness metric provides more accurate performance results by considering the type of bug detected by participants. For instance, participants who identified more important bugs get better performance results compared to those who identified non-critical bugs. For this comparison, we calculated the average performance of

participants using both metrics. Furthermore, average performance of participants in identifying total number of bugs helped to find the relationship between the scores obtained from effectiveness metric and number of bugs detected by the participants. In contrast, average performance of participants in detecting different categories of bugs helped the researcher to identify the relationship between the performance of participants and their performance in identifying different levels of defects using the gamified effectiveness metric.

Moreover, gamified validity and gamified f score metrics provide more accurate evaluation result considering different level of bugs while the existing validity and f score metrics do not consider the importance of bugs identified by testers. Table 4 presents the performance evaluation of participants using validity metric while Table 5 shows the performance evaluating using gamified validity metric. Furthermore, as discussed earlier, for decision making purposes, the combination measures of effectiveness and validity will help to evaluate the performance of participants more accurately. Table 6 represents the participants' performance evaluation using the f score metric. However, in order to provide more accurate and fairer results, gamified f score metric was used. Table 7 represent the results of participants' performance using gamified f score metric.

	Performance	Score (number)	Total Bugs (%)
Effectiveness	**Low**	0.1235	12.35 %
	Medium	0.2941	29.41%
	High	0.6039	60.39%

Table 2. Performance evaluation using effectiveness metric

	Performance	Score (number)	Easy (%)	Medium (%)	Hard (%)
Gamified Effectiveness	**Low**	0.0797	14.28%	15.47%	0.95%
	Medium	0.2893	30.35%	28.57%	28.57%
	High	0.6055	66.66%	50%	64.44%

Table 3. Performance evaluation using gamified effectiveness metric

	Performance	Score (number)	Total Bugs (%)	Invalid bugs (number)
Validity	**Low**	0.1108	12.35 %	2.5
	Medium	0.2727	29.41%	1.42
	High	0.5692	60.3%	1

Table 4. Performance evaluation using validity metric

	Performance	Score (number)	Easy (%)	Medium (%)	Hard (%)	Invalid bugs
Gamified Validity	**Low**	0.0740	14.28%	15.47%	0.95%	2.5
	Medium	0.2764	30.35%	28.57%	28.57%	1.42
	High	0.5666	66.66%	50%	64.44%	1

Table 5. Performance evaluation using gamified validity metric

	Performance	Score (number)	Total Bugs (%)	Invalid bug(number)
F score	**Low**	0.1166	12.35 %	2.5
	Medium	0.2829	29.41%	1.42
	High	0.5857	60.3%	1

Table 6. Performance evaluation using f score metric

	Performance	Score (number)	Easy (%)	Medium (%)	Hard (%)	Invalid bug(number)
Gamified F score	**Low**	0.0767	14.28%	15.47%	0.95%	2.5
	Medium	0.2826	30.35%	28.57%	28.57%	1.42
	High	0.5845	66.66%	50%	64.44%	1

Table 7. Performance evaluation using gamified f score metric

5.2.2 Effectiveness of Software Testers' Performance Based on the Level of Difficulty

In this section, we study the effect of gamification on the performance of the participants based on the level of task difficulty. Table 8 presents the performance of all participants in the given tasks. In order to calculate the performance of the participants, we use the Effectiveness, validity and f score metrics respectively. It is important to note that the proposed gamified metrics could evaluate the overall performance of participants when considering all levels of bugs. However, in this case, we are evaluating the performance of testers focusing on a specific category of bugs. Thus, the existing metrics will be used to evaluate the participants' performance. In order to evaluate the performance of the participants we use the following metrics:

1. Effectiveness = vdf / vdt
2. Validity= (tp / (tp + fp))
3. F score = 2 * ((Validity * Effectiveness) / (Validity + Effectiveness))

Results suggest that the performance of participants in detecting easy bugs in the gamified information system testing platform was higher compared to other levels categories. The average performance of all 25 participants in performing testing activity to detect bugs categorised as easy using f score metric was 23.65% while their average performance for detecting medium and hard bugs were 18.9% and 12.8% respectively. In order to calculate these, measures that were presented in the information retrieval domain were used (Baeza-Yates & Ribeiro-Neto, 1999). However, these measures have been partially adopted by the usability community (Hartson et al., 2001) and software engineering community (Mäntylä & Itkonen, 2013).

Measures	Easy (Task 1)	Easy (Task 2)	Average Performance for Easy Task 1 & 2	Medium (Task 1)	Medium (Task 2)	Average Performance for Medium Task 1 & 2	Hard (Task 1)	Hard (Task 2)	Average Performance for Hard Task 1 & 2
Effectiveness	24.66%	24%	**24.33%**	8%	30%	**19%**	4%	22.4%	**13.2%**
Validity	23.06%	23.33	**23.19%**	8%	29.8%	**18.9%**	4%	21.06%	**12.53%**
F score	23.7%	23.6%	**23.65%**	8%	29.8%	**18.9%**	4%	21.6%	**12.8%**

Table 8. Effectiveness of software testers' performance based on the level of difficulty

6 Discussion and Limitation

In this paper, results were provided after evaluating the final gamified information system testing platform through the evaluation session conducted with 25 undergraduate computing students. Due to higher expectation or different view in relation to evaluation of gamified information system testing by a larger group of professional software testers, the results obtained may impact the outcome of this study. Additionally, in order for students to show interest in participating in the focus group session, the duration of the session had to be less than an hour period. Thus, time pressure may be the main factor that affected the participants' performances in this study and resulted in lower performance rate by the participants. Moreover, in the given questionnaire, students agreed that the time pressure could be a factor to affect the quality and performance of software testers. In this study, researchers have tried to choose participants who had knowledge in both software testing (mainly unit testing) and software development. Finally, researcher discovered that the proposed metric is beneficial when evaluating the performance of software testers in combination of different levels of bugs (by considering the importance of bugs identified by software testers). However, in order to evaluate the performance of software testers for a specific category of bugs, the proposed metric may not be beneficial to evaluate the performance of software testers.

7 Conclusion and Future Work

In this paper, the topics of gamification, information system testing, and effect of time restriction on performance were discussed. Results suggested that majority of participants agreed to the fact that time pressure may compromise the performance of software testers impacting on the information system testing efficacy. The quantitative results suggested that the performance of software testers was affected by the time pressure introduced during the prototype evaluation. In addition, a set of new metrics were proposed to better capture the performance of software testers. It has been demonstrated that these metrics are able to fairly distribute the scores to reflect on the types of bugs being reported. Further work will include evaluating the use of gamified software-testing platform, the performance of software testers working in teams versus individually. Additionally, researcher plans to investigate the importance of introducing the proposed metrics for evaluating the performance of software testers in the gamified software-testing platform prior to conducting the testing activity. This then will help to identify if the new metrics could be beneficial in assisting the testers to detect

higher priority bugs within the software. Finally, researcher also plan to investigate the amount of time required for information system testing tasks. This may help to identify if this can lead to higher level of performance efficiency.

References

Agrawal, M., & Chari, K. (2007). Software effort, quality, and cycle time: A study of CMM level 5 projects. *IEEE Transactions on Software Engineering, 33*(3).

Alrmuny, D. Z. (2014). Open problems in software test coverage. *Lecture Notes on Software Engineering, 2*(1), 121.

Alsawaier, R. S. (2018). The effect of gamification on motivation and engagement. *The International Journal of Information and Learning Technology, 35*(1), 56-79.

Arai, S., Sakamoto, K., Washizaki, H., & Fukazawa, Y. (2014). *A gamified tool for motivating developers to remove warnings of bug pattern tools.* Paper presented at the 6th International Workshop on Empirical Software Engineering in Practice (IWESEP), 2014 .

Arnarsson, D., & Jóhannesson, Í. H. (2015). *Improving Unit Testing Practices with the Use of Gamification*. Chalmers University of Technology, Gothenburg, Sweden.

Austin, R. D. (2001). The effects of time pressure on quality in software development: An agency model. *Information systems research, 12*(2), 195-207.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval* (Vol. 463): ACM press New York.

Baeza-Yates, R., & Ribeiro, B. d. A. N. (2011). *Modern information retrieval*: New York: ACM Press; Harlow, England: Addison-Wesley.

Barata, G., Gama, S., Jorge, J., & Gonçalves, D. (2013). *Engaging engineering students with gamification.* Paper presented at the 5th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES), 2013.

Bartol, K. M., Durham, C. C., & Poon, J. M. (2001). Influence of performance evaluation rating segmentation on motivation and fairness perceptions. *Journal of applied psychology, 86*(6), 1106.

Blum, B. I. (1992). *Software engineering: a holistic view*: Oxford University Press, Inc.

Briand, L. C. (2007). A critical analysis of empirical research in software testing. Paper presented at the *First International Symposium on Empirical Software Engineering and Measurement* (ESEM 2007), IEEE.

Chen, N., & Kim, S. (2012). *Puzzle-based automatic testing: Bringing humans into the loop by solving puzzles.* Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2012.

Connolly, T. M., Boyle, E. A., MacArthur, E., Hainey, T., & Boyle, J. M. (2012). A systematic literature review of empirical evidence on computer games and serious games. *Computers & Education, 59*(2), 661-686.

De Freitas, A. A., & de Freitas, M. M. (2013). Classroom Live: a software-assisted gamification tool. *Computer Science Education, 23*(2), 186-206.

de Jesus, G. M., Ferrari, F. C., de Paula Porto, D., & Fabbri, S. C. P. F. (2018). *Gamification in Software Testing: A Characterization Study.* Paper presented at the Proceedings of the III Brazilian Symposium on Systematic and Automated Software Testing.

de Sousa Borges, S., Durelli, V. H., Reis, H. M., & Isotani, S. (2014). *A systematic mapping on gamification applied to education.* Paper presented at the Proceedings of the 29th Annual ACM Symposium on Applied Computing.

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). *From game design elements to gamefulness: defining gamification.* Paper presented at the Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments.

Djaouti, D., Alvarez, J., & Jessel, J.-P. (2011). Classifying serious games: the G/P/S model. In *Handbook of research on improving learning and motivation through educational games: Multidisciplinary approaches* (pp. 118-136): IGI Global.

Dunsmore, A., Roper, M., & Wood, M. (2003). The development and evaluation of three diverse techniques for object-oriented code inspection. *IEEE Transactions on Software Engineering, 29*(8), 677-686.

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

García, F., Pedreira, O., Piattini, M., Cerdeira-Pena, A., & Penabad, M. (2017). A framework for gamification in software engineering. *Journal of Systems and Software, 132*, 21-40.

Gross, E., & Etzioni, A. (1985). *Organizations in society*: Prentice-Hall.

Hamari, J., Koivisto, J., & Sarsa, H. (2014). *Does gamification work?--a literature review of empirical studies on gamification.* Paper presented at the 2014 47th Hawaii international conference on system sciences (HICSS).

Hartson, H. R., Andre, T. S., & Williges, R. C. (2001). Criteria for evaluating usability evaluation methods. *International journal of human-computer interaction, 13*(4), 373-410.

Johansson, M., & Ivarsson, E. (2014). *An experiment on the effectiveness of unit testing when introducing gamification.* PhD thesis, Master's thesis, Chalmers University of Technology (June),

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

Kazhamiakin, R., Marconi, A., Perillo, M., Pistore, M., Valetto, G., Piras, L., . . . Perri, N. (2015). *Using gamification to incentivize sustainable urban mobility.* Paper presented at the IEEE First International Smart Cities Conference (ISC2), 2015.

Kumar, J. (2013). Gamification at work: Designing engaging business software. Paper presented at the *International conference of design, user experience, and usability*, Berlin Heidelberg, 2013.

Lau, C. M., Wong, K. M., & Eggleton, I. R. (2008). Fairness of performance evaluation procedures and job satisfaction: The role of outcome-based and non-outcome-based effects. *Accounting and Business Research, 38*(2), 121-135.

Leitner, A., Ciupa, I., Meyer, B., & Howard, M. (2007). *Reconciling manual and automated testing: The autotest experience.* Paper presented at the 40th Annual Hawaii International Conference on System Sciences, 2007. HICSS 2007..

Luo, L. (2001). Software testing techniques. *Institute for software research international Carnegie mellon university Pittsburgh, PA, 15232*(1-19), 19.

Lyubomirsky, S., & Ross, L. (1997). Hedonic consequences of social comparison: a contrast of happy and unhappy people. *Journal of personality and social psychology, 73*(6), 1141.

Mäntylä, M. V., & Itkonen, J. (2013). More testers–The effect of crowd size and time restriction in software testing. *Information and Software Technology, 55*(6), 986-1003.

Mäntylä, M. V., & Smolander, K. (2016). *Gamification of Software Testing-An MLR.* Paper presented at the International Conference on Product-Focused Software Process Improvement.

Marciniak, J. J. (1994). *Encyclopedia of software engineering (vol. 1 AN)*: Wiley-Interscience.

McDaniel, L. S. (1990). The effects of time pressure and audit program structure on audit performance. *Journal of Accounting Research*, 267-285.

Memar, N., Krishna, A., McMeekin, D. A., & Tan, T. (2017). Gamification of Information System Testing-Design Consideration through Focus Group Discussion. Paper presented at the *26th international Conference on Information Systems Development*, Sep 6, 2017, Larnaca, Cyprus.

Memar, N., Krishna, A., McMeekin, D. A., & Tan, T. (2018). Gamifying Information System Testing–Qualitative Validation through Focus Group Discussion. Paper presented at the *27th international Conference on Information Systems Development*, Aug 9, 2018, Larnaca, Sweden.

Nah, F. F.-H., Telaprolu, V. R., Rallapalli, S., & Venkata, P. R. (2013). *Gamification of education using computer games.* Paper presented at the International Conference on Human Interface and the Management of Information.

Nan, N., & Harter, D. E. (2009). Impact of budget and schedule pressure on software development cycle time and effort. *IEEE Transactions on Software Engineering, 35*(5), 624-637.

Pedreira, O., García, F., Brisaboa, N., & Piattini, M. (2015). Gamification in software engineering–A systematic mapping. *Information and Software Technology, 57*, 157-168.

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

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

Shah, H., & Harrold, M. J. (2010). *Studying human and social aspects of testing in a service-based software company: case study.* Paper presented at the Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of software Engineering.

Sholihin, M., & Pike, R. (2009). Fairness in performance evaluation and its behavioural consequences. *Accounting and Business Research, 39*(4), 397-413.

Singer, L., & Schneider, K. (2012). *It was a bit of a race: Gamification of version control.* Paper presented at the 2nd International Workshop on Games and Software Engineering (GAS), 2012

Topi, H., Valacich, J. S., & Hoffer, J. A. (2005). The effects of task complexity and time availability limitations on human performance in database query tasks. *International Journal of Human-Computer Studies, 62*(3), 349-379.

Valett, J. D., & McGarry, F. E. (1989). A summary of software measurement experiences in the software engineering laboratory. *Journal of Systems and Software, 9*(2), 137-148.

Van Rijsbergen, C. J. (1986). A non-classical logic for information retrieval. *The computer journal, 29*(6), 481-485.

Von Ahn, L. (2013). *Duolingo: learn a language for free while helping to translate the web.* Paper presented at the Proceedings of the 2013 international conference on Intelligent user interfaces.

Von Ahn, L., Maurer, B., McMillen, C., Abraham, D., & Blum, M. (2008). recaptcha: Human-based character recognition via web security measures. *Science, 321*(5895), 1465-1468.

Whyte, E. M., Smyth, J. M., & Scherf, K. S. (2015). Designing serious game interventions for individuals with autism. *Journal of autism and developmental disorders, 45*(12), 3820-3831.

Yerkes, R. M., & Dodson, J. D. (1908). The relation of strength of stimulus to rapidity of habit-formation. *Journal of comparative neurology and psychology, 18*(5), 459-482.

doi: https://doi.org/10.3127/ajis.v24i0.2179