Gamification of Information System Testing - Design Consideration through focus group discussion

Navid Memar *navid.memar@postgrad.curtin.edu.au*
Department of Computing, Curtin University
Bentley 6102, Australia

Aneesh Krishna *A.Krishna@curtin.edu.au*
Department of Computing, Curtin University
Bentley 6102, Australia

David A McMeekin *D.McMeekin@curtin.edu.au*
Department of Spatial Sciences, Curtin University
Bentley 6102, Australia

Tele Tan *t.tan@curtin.edu.au*
Department of Mechanical Engineering, Curtin University
Bentley 6102, Australia

Abstract

This paper presents the design consideration of gamifying Information System Testing through results obtained from multiple focus group sessions with software developers and testers. The intention is to provide new testing strategies and tools that facilitates Information System testing to be more efficient while providing the tester with an engaging and rewarding test environment. Gamification may be a solution to raise testers' engagement, as it can potentially remedy the high degree of repetition and ensuing boredom for a tester in the testing phase. Findings, suggest that gamification may be an element to advance testing efficiency. Moreover, system and task knowledge, the objectives of the actual system, time constraints and required tools for performing testing tasks are essential for efficient testing performance. Participants stated that points, levels, meaningful gifts, progressions, provision of choice and feedback are the main elements a gamified testing platform must contain.

1. Introduction

Day by day we are becoming more reliant on software to manage our daily activities. Testing is a vital activity in software development process and the purpose is to ensure that the software is bug free and of high quality for its intended purpose [37]. As software innovation rapidly progresses, the number of software failures may unavoidably increase and if the failure is massive, it can destroy a company's future [7]. One way to reduce the number of software failures is through robust testing which can be performed either manually or automatically. Ideally, both types of testing are executed. These two testing approaches can be seen as complementary; automated testing is capable of performing a large number of test scenarios in a relatively short period of time whereas manual testing uses the tester's knowledge to capture deep or specific cases which automated testings may fail to detect [26]. In addition, the software testing process is a repetitive and boring task and may not be ideal for a large number of software developers to perform [30].

Lack of testing may result in problems and many software problems are due to bad testings. As an example, the problem occurred in the New Derivation Trading System, related to Tokyo Stock Exchange's derivative products on February 8, 2008. As the result of the problem with the system, the trading system operation was suspended for several months. The reports showed

the failure was due to an initialization error with the memory within the server [25]. The risk can be higher by not identifying hazardous software failures and allocating suitable safety requirements [21]. Recent studies have demonstrated that software bugs may cause major financial concerns which would be critical for oranizations [19]. In early 2016, HSBC stated that a complex technical issue caused a major IT outage in their organization [19]. Moreover, in August 2015, HSBC failed to process 275,000 individual payments due to a bug in their electronic payment system [19].

Another example of catastropic software failure involved the NASA Mars Polar Lander in January 1999. In this instance, the cause of the accident was identified as software error which caused the premature shutdown of the decent engines of the lander [21]. The loss of the Mars Polar Lander (MPL) provides an example of bad software verification and testing which were the main factors of this incident. On 7th October 2008 an in-flight upset occurred on an Airbus A330-303 enroute to Perth from Singapore at the cruising altitude of 37,000ft. Several passengers and crew on borad were injured. Investigations revealed that one of the Interial Reference Units (ADIRUs) started to output intermittently incorrect values of vital flight prameters to the rest of aircraft subsystems [15].

Clearly the demand for software testing is on the increase which mirrors the rapid growth in software for industrial and commercial purposes. However, graduates and candidates that are looking for positions in the Technology sector are more inclined to lean towards software development and they do not show much interest towards software testing [11]. With the increasing number of IT industries and complexity of delivered software, organizations require more professional software testers [11].

Recently there has been a high level of interest in the use of game elements in non-gaming software applications with the aim, to raise engagement and motivation [9, 16]. Gamification can be implemented to transform difficult testing tasks into an engaging and motivating experience that taps into the gamer's creativity with the possibility to bring about the detection of defects that may not have been detected in the traditional manner [37].

Gamification is becoming a growing field of interest in many domains such as Testing [8], Character recognition [44], Education and Academia [23] [38] Language Translation [43], Version Control [42] and many others. By applying gamification, the level of user engagement and productivity may be increased [5].

This paper focuses on creating new testing strategies and tools that can help software testing to be more efficient while providing an engaging and rewarding environment for the tester. To reach this goal, the current monotonous nature of testing will be addressed through the gamification of testing. In this paper, a web based game engine prototype is introduced that implements software testing gamification. It is in an early stage of prototyping, but the platform is being used to evaluate the significance of having game elements involved in the software testing process. The current platform has been used to obtain useful information from experienced developers and testers throughout multiple focus group sessions. Their feedback provided excellent input into the features to implement in the finalised platform.

2. Background

The aim of software testing is to find errors and to determine the software quality and recent software problems are due to lack of testing [25]. Software testing also helps provide assurance that the software is working correctly and according to specification. In other words, software testing can be used to verify and validate the software quality. Software testing is an important part of software engineering. Software testing takes 40 to 50% of development effort and this practice could also use more effort for more complex systems which require higher level of quality [29]. In every stage of the software life cycle, testing is involved. However, testing in each part of software development is different [29]. Software testing activity could be seen from two views of technical and a socio-technical view [33]. It is good to know that software testing is a social-technical even though most parts of software engineering is purely technical [31]. Software testing is usually tedious, monotonous and boring practice [41] [22]and is usually time- consuming, difficult and often inadequate [1]. With having systems more complex and

the increased usage of IT as well as high software critical systems, software testing difficulty is expected to rise in future together with a rise in total costs of low software quality [4]. By taking the socio-technical aspect into consideration, the skill and motivation of software testers are vital to increase software quality and for the developer to be successful in software testing [4].

2.1. Gamification

Gamification uses game-based elements: mechanics, aesthetics, and game thinking in non-game contexts with the aim of gamification to provide a learning environment, solving problems, engaging and motivating users [10] [40]. Any process that impacts employees could be gamified to improve engagement or experience, from selection and recruitment to training and performance [6].

Gamification consists of applying elements of gamefulness, gameful interaction and gameful design with a unique intention in mind [13]. In recent years, gamification has been an active research topic and a subject of much interest as a means of supporting user engagement and enhancing positive patterns in service use, such as increasing user activity, social interaction, or quality and productivity of action [20]. As a result of gamification, people are engaged in tasks and specific problems are solved [24]. Gamification has recently been successful in website marketing to create loyalty, brand awareness and effective marketing engagement [32]. It is possible that gamifying software testing may lead to more successful software testing through the identifying more defects earlier in the development life cycle.

Many digital games use different techniques and resources and these techniques have elements to motivate users and challenge them to solve problems and complete different stages and tasks. In gamification approaches, these elements are not the centre of the system, but have the purpose of motivating users to use it [36]. Gamification has been widely used in recent years in different domains of which some are described. Gamification may be a solution to increase tester's engagement in software testing. Testing contains much repetition and may seem boring. To make this important part of software engineering more attractive, gamification may be a method to increase testing quality and make it more engaging for the human software tester.

2.2. Education

Gamification in Education refers to the use of game elements and game thinking for educational purposes and academic development in both formal and informal aspects. Denny [12] examined the usefulness of badges to increase participants' motivation in a study of an online multiple-choice questionnaire (MCQ). Results showed a high positive impact on the number of students' contribution, without impacting their corresponding quality. Denny [12] stated that students showed much interest in having badges available in their user interface and being able to earn them.

Dominquez [14] applied a gamification plugin for a well-known e-learning platform (Blackboard). Results of the experiment showed both positive and negative points of using gamification. Their findings revealed that better scores in practical assignments and in the overall score was achieved by those who completed the gamified experience. However, their study also showed that those same students did not perform well in their written assignments and had a lower participation rate in class activities, even though their motivation was high [14].

Goehle [18] experimented using video game mechanics with an online homework program called WeBWorK in order to increase student engagement with online mathematics homework. Goehle used few gamification elements: experience points based on student progress through individual tasks, levels for reaching specific milestones and a progress barn, achievements and rewards as of extra point. Even though student progress was satisfactory, the authors were unable to conclusively decide if gamification had any effect on students' performances in the course [18, 40].

Li [27] implemented a gamified tutorial system (GamiCAD) to allow new AutoCAD users to study AutoCAD through a gamified process. Gamified elements used in this system were

missions, scoring, levels, time limit, mini games and rewards. Results indicated a better engagement and performance between the users [27] [40].

2.3. Crowdsourcing

Liu [28] developed a mobile crowdsourcing application called UbiAsk for the purpose of image based social search across languages. Gamification elements were used in the application to encourage participants to translate images. The researchers discovered good participation and response by the users. However, it was unclear how the results compared to the non-gamified version of the application. In addition, results indicated gamification use did not reach statistical significance [28].

Witt [47] examined what motivates users who participate in an online idea competition and what negative results could the game mechanism have on motivation. Game points and social points were used in the systems. Game points were awarded for completion of actions and social points for engagement in social behaviours (rating, commenting on the users' idea etc).

The questionnaire's results regarding the game mechanics fell into the "Neither Agree nor Disagree" category. The authors speculated that the system design, together with the leader board presentation, provided these mixed results [40].

2.4. Research

Rapp, Marcengo [35] implemented a gamified application to help raise the amount of valid response from users in a field study. WantEat was the application's name which was fit for a study taken around a cheese festival. Users received points based on actions such as providing feedbacks and tasting cheese with other contributors. Their progress was tracked in a leaderboard and achievements were unlocked and additional rewards earned by users. Results showed users rated application ease of use, usefulness and high engagement of the application. Nevertheless, participants were not motivated to communicate more with new people or leave comments on other people's reviews. In addition to that, users continuously used the application even when they reached the maximum point level [40]. Their conclusion, it was not clear whether users were motivated to use the application due to the game element components or not [40].

3. Methodology

This section explains the system's implementation methods and the way game elements will be used in the chosen software testing area. However, it is essential to first choose a relative methodology in order to obtain optimal results. Therefore, the design science method based on the regulative cycle framework proposed by Wieringa [46] has been selected. This methodology places emphasis on the linkage of knowledge and practice to design for utility whilst acquiring better scientific knowledge and results. Figure 1 represents the regulative cycle suggested by Wieringa[46] and Figure 2 contains an explanation on how practical problems will be addressed in different stages.

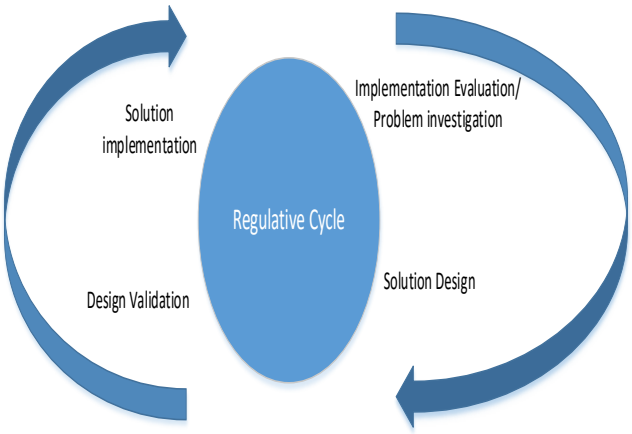

Fig 1. The regulative cycle [46]

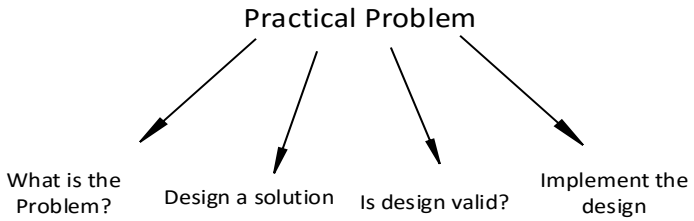

Fig 2. Breaking down practical problems into sub-problems

3.1. Problem investigation (Gaming and Testing)

Problem investigation involves knowledge and an understanding of existing problems. A detailed explanation of existing problems was given in the literature review section. It should be noted that lack of testing may cause harm, and recent major issues with software have largely been due to unscrupulous software testings. Clearly the demand for software testing is on the increase, which mirrors the rapid growth of software for industrial and commercial purposes. However, graduates and candidates that are looking for positions in the Information Technology industry tend to concentrate their efforts on software development, with less consideration of opportunities in software testing. The increasing importance of IT industries and the complexity of delivered products and systems organizations have led to an increased demand for professional software testers[11].

3.2. Solution- driven investigation

Technology is rapidly growing and is in search of problems to be resolved. The main goal of this research is to study and investigate the relationship between gamification and the sustainability of the testing environment by understanding software testers' behaviour to improve software testing quality and provide an engaging environment for testers. Gamification is the main method to be studied and analysed as potentially beneficial element to solve some current issues of software testing. By applying game elements (as discussed earlier in gamification elements section) to the proposed application, the possible advantages and disadvantages of using this method in the software testing area may gauged. Gamification may be a solution to increase testers' engagement in software testing, as it can potentially remedy the high degree of repetition and the ensuing boredom of a tester in the testing phase. Thus, to make this important part of software engineering more attractive, gamification may be a method to improve the quality of testing by making testing a more stimulating experience for the human software tester by reducing testers' boredom.

3.3. Proposed Solution: Mapping Game Elements with Design Elements of Serious Games

To study and understand what game elements suit the needs of this study, many papers were reviewed. Clearly the focus is to increase the engagement and raise the productivity level of testers in parallel with growth in the quality of the testing practice. Whyte et al. [45] suggested core principles of serious game design and explains clearly that these would enhance motivation to play. In this practice, those core principles were adopted and helped to identify useful elements to be applied to this study's prototype.

The key element to increase the motivation of players in serious game design is story line [45]. Story line is a helpful element to grow the motivation for learning when learning opportunities are combined directly with the story content [2, 17]. Therefore, it was decided to apply this element to add to a higher level of motivation, engagement and quality.

The second game element is goals directed learning around targeted skills [45]. This practice is with the combination of primary end goals and intermediate incremental goals which provide challenges and progress [45]. Both medium and long-term goals have been applied to the current system to provide motivation and goals for the players.

Thirdly, feedback and awards play a very crucial role in shaping behaviour in serious games as players intend to work continuously to achieve certain goals [45]. Multiple feedback engines were injected into the current system to provide real time as well as post exercise feedback to players. Increasing levels of difficulty and individuation also play an important role in serious games. Providing challenging, but achievable goals in a supportive environment helps to increase competence for certain skills [34]. Serious games should endeavour to attain a challenging, but achievable level of difficulty for players. Thus, it follows that these games should not be so difficult so as to discourage players from attempting to complete the game, but also not so easy that players are demotivated from learning new skills [45]. In order to achieve this, different badges and levels were implemented in the current system and players were enabled to unlock each after attaining certain points upon completing a task successfully.
Provision of choice is another element that increases motivation and enjoyment in serious games [39]. Allowing individuals to have choice over some aspects of the game environment helps players to maintain a sense of autonomy and control over their learning experience [34]. In the current system, players have the ability to accept or reject a given task after it is assigned. Upon rejecting the task, it is automatically sent to the next available player for review.

Focus groups were the main method chosen by the researcher to determine the main elements to be applied in the system to gamify the software testing process. Various focus group sessions were conducted with developers and testers to understand and identify the key gamification elements in an effort to give software testers a more interactive tool and environment to test the production system. In the next section, a detailed explanation of how this method was applied, has been described.

The aim is to study and understand if these game core elements and design elements may help significantly to increase learning and motivation.

4. Results and Analysis

Conducting various focus groups with developers and testers helped the researcher to ascertain key factors to optimize a system which gamifies the software testing process.

In the following sections, the main themes and findings are described that emerged from the focus group sessions conducted. Written questionnaires and recorded focus-group discussions were used in gathering this information. The process of analysis began by reviewing the list of findings in all focus group sessions and the recorded focus-group discussions. Key factors were identified and were coded with the number of similar responses associated with each factor in order to provide a precise result. Those numbers then were converted into graphs to represent the findings in a simplified method.

In total 20 students attended the sessions and out of them 70% had both a Software Development and Software Testing background, 25% of them had a Software Development

background and 5% had only a Software Testing background. These findings can be summarized under the following headings:

4.1. Gamification encourages software testers

In the questionnaire, a question asked whether gamification could be an element to rectify the issues of disengagement for software testers. To support their answers further, participants also explained their point of view by detailed explanation.

Most participants stated clearly that gamification may be an element to improve the testing efficiency and to rectify the issues in software testing. Two responses are listed below:

- "Absolutely! However, the environment would be designed in such a way that is very intuitive to use, provides results, and gives testers something to work towards."
- "Yes, gamification makes people get more interested and competitive."

On the other hand, another participant argued that although gamification could avoid the repetitive nature of software testing, other issues might arise. The participant has supported his statement that real gifts or learning new skills may be beneficial for the testers but testers would not test the code simply for recreation in terms of the gamification aspect.

4.2. Requirement and design documentation is a vital information required for testers

To identify the important information required for software testers to perform a testing task efficiently, participants were given related question and their responses have been coded into four main factors (requirement & design documentation, tools, time and knowledge).

Figure 3 represents the findings obtained from the various focus group sessions. From Figure 3, it is clear that requirements and design documentation as well as knowledge are the key factors. The results from the focus group sessions suggests that system and task knowledge, the objectives of the actual system, time constraints and required tools for performing the task are essential to have an efficient testing performance. The responses have been coded and categorized into four sections shown in the figure 3.

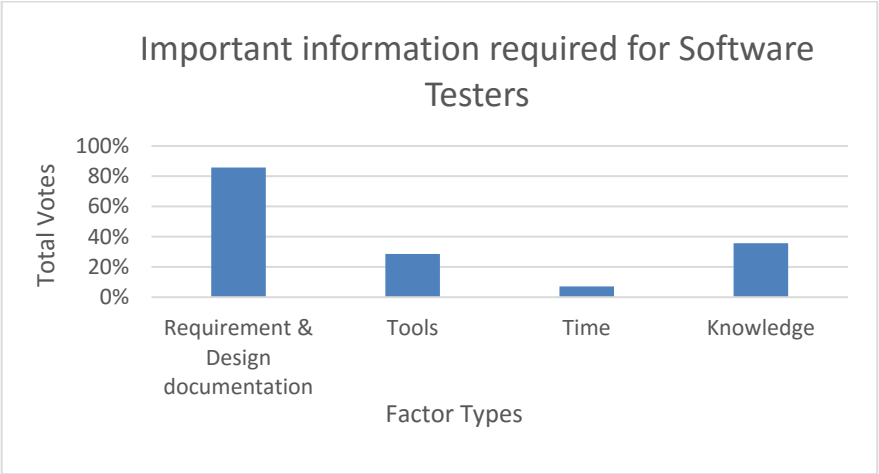

Fig 3. Factors required for Software Testers

4.3. Number of testers relate to the quality of testing

The results suggest that participants were agreed on the fact that having more testers can increase the quality of testing. Some of the responses were as such:

- "Theoretically the more testers there are, the more bugs are going to be identified. Two brains are better than one."

- "Different individuals will approach problems in different ways and will likely find different bugs. No matter how good a tester is, it is always better to have multiple testers."
- "More numbers of testers means more numbers of results. You get more results from different types of users as well as different level of expertise."
- "The more the testers, the better quality."

Some also have argued that even though more testers could improve the testing performance, it could also complicate the testing process. The following is an example to support this idea:

- "In some cases, the number of testers positively influences the result of the testing. For instance, a greater number of testers can lead to a greater ability to recognise defects in the system. However, in other cases, a greater number of testers can complicate the testing process."

4.4. Essential game elements

Figure 4 presents motivational factors stated by participants. Under such circumstances, it is clear that all five categories are essential for the proposed system. Game elements such as, points, levels, real gifts, progressions, character development, provision of choice and feedback were the main factors stated by participants which, after precise consideration, have been coded into the main five design elements suggested by Elisabeth[45]. Among these 5 categories, participants voted the most for feedback and rewards shape learning element. Kay Berkling [3]; also stated that students are mostly looking for elements such as attractive design, intuitive control, task overview, feedback, communication section, comparison tool, leader board, points, badges and levels in a gamified platform.

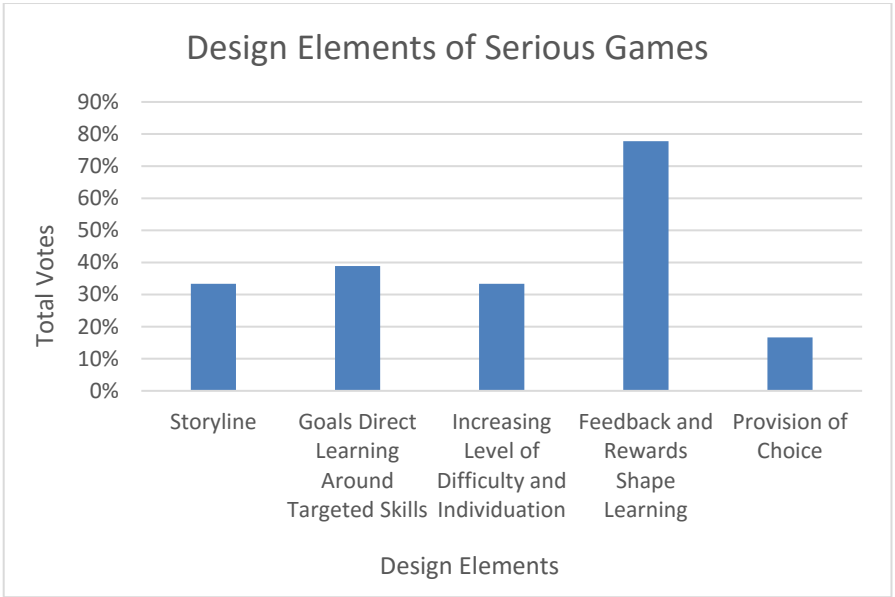

Fig 4. Design elements and motivation factors

5. Conclusion and Future Work

In this paper a brief background of software testing and selected method was discussed. In addition, the proposed idea of how gamification could engage software testers and motivate them was discussed. Throughout the focus group sessions and results obtained from those sessions, proposed game elements have been identified. The next step to this research is to design a gamified platform for testers based on the identified game elements. Furthermore, an

evaluation of the final platform may identify if gamification can be a factor to increase the quality of testing and increase the engagement of testers.

Note: The study obtained ethics approval from office of Research and Development, Curtin University, Australia with Ethics approval number: **RDSE-76-15** and participants gave informed consent before participating in focus group sessions.

References

1. Alrmuny, D.Z., Open Problems in Software Test Coverage. Lecture Notes on Software Engineering, 2014. 2(1): p. 121.
2. Baranowski, T., R. Buday, D.I. Thompson, and J. Baranowski, Playing for real: video games and stories for health-related behavior change. American journal of preventive medicine, 2008. 34(1): p. 74-82. e10.
3. Berkling, K. and C. Thomas. Gamification of a Software Engineering course and a detailed analysis of the factors that lead to it's failure. in Interactive Collaborative Learning (ICL), 2013 International Conference on. 2013. IEEE.
4. Bertolino, A. Software testing research: Achievements, challenges, dreams. in 2007 Future of Software Engineering. 2007. IEEE Computer Society.
5. Buckley, P. and E. Doyle, Gamification and student motivation. Interactive Learning Environments, 2016. 24(6): p. 1162-1175.
6. Callan, R.C., K.N. Bauer, and R.N. Landers, How to avoid the dark side of gamification: Ten business scenarios and their unintended consequences, in Gamification in education and business. 2015, Springer. p. 553-568.
7. Charette, R.N., Why software fails [software failure]. Ieee Spectrum, 2005. 42(9): p. 42-49.
8. Chen, N. and S. Kim. Puzzle-based automatic testing: Bringing humans into the loop by solving puzzles. in Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering. 2012. ACM.
9. da Rocha Seixas, L., A.S. Gomes, and I.J. de Melo Filho, Effectiveness of gamification in the engagement of students. Computers in Human Behavior, 2016. 58: p. 48-63.
10. de Sousa Borges, S., V.H. Durelli, H.M. Reis, and S. Isotani. A systematic mapping on gamification applied to education. in Proceedings of the 29th Annual ACM Symposium on Applied Computing. 2014. ACM.
11. Deak, A., T. Stålhane, and D. Cruzes, Factors influencing the choice of a career in software testing among norwegian students. Software Engineering, 2013: p. 796.
12. Denny, P. The effect of virtual achievements on student engagement. in Proceedings of the SIGCHI conference on human factors in computing systems. 2013. ACM.
13. Deterding, S., 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. 2011. ACM.
14. Domínguez, A., J. Saenz-de-Navarrete, L. De-Marcos, L. Fernández-Sanz, C. Pagés, and J.-J. Martínez-Herráiz, Gamifying learning experiences: Practical implications and outcomes. Computers & Education, 2013. 63: p. 380-392.
15. Favarò, F.M., D.W. Jackson, J.H. Saleh, and D.N. Mavris, Software contributions to aircraft adverse events: Case studies and analyses of recurrent accident patterns and failure mechanisms. Reliability Engineering & System Safety, 2013. 113: p. 131-142.
16. Fitz-Walter, Z., D. Tjondronegoro, and P. Wyeth. A gamified mobile application for engaging new students at university orientation. in Proceedings of the 24th Australian Computer-Human Interaction Conference. 2012. ACM.
17. Garris, R., R. Ahlers, and J.E. Driskell, Games, motivation, and learning: A research and practice model. Simulation & gaming, 2002. 33(4): p. 441-467.
18. Goehle, G., Gamification and web-based homework. Primus, 2013. 23(3): p. 234-246.

19. Goyal, A. and N. Sardana, Bug Handling in Service Sector Software, in Applying Predictive Analytics Within the Service Sector. 2017, IGI Global. p. 54-73.
20. Hamari, J., J. Koivisto, and H. Sarsa. Does gamification work?--a literature review of empirical studies on gamification. in System Sciences (HICSS), 2014 47th Hawaii International Conference on. 2014. IEEE.
21. Hawkins, R., I. Habli, and T. Kelly. The principles of software safety assurance. in 31st International System Safety Conference, Boston, Massachusetts USA. 2013.
22. Kanij, T., R. Merkel, and J. Grundy. An empirical study of the effects of personality on software testing. in Software Engineering Education and Training (CSEE&T), 2013 IEEE 26th Conference on. 2013. IEEE.
23. Kapp, K.M., The gamification of learning and instruction: game-based methods and strategies for training and education. 2012: John Wiley & Sons.
24. Kim, B., . Gamification. Library Technology Reports, 2015. 51(2): p. 10-18.
25. Kurokawa, T. and M. Shinagawa, Technical trends and challenges of software testing. Science & Technology Trends, 2008. 10: p. 34-45.
26. Leitner, A., I. Ciupa, B. Meyer, and M. Howard. Reconciling manual and automated testing: The autotest experience. in System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on. 2007. IEEE.
27. Li, W., T. Grossman, and G. Fitzmaurice. GamiCAD: a gamified tutorial system for first time autocad users. in Proceedings of the 25th annual ACM symposium on User interface software and technology. 2012. ACM.
28. Liu, Y., T. Alexandrova, and T. Nakajima. Gamifying intelligent environments. in Proceedings of the 2011 international ACM workshop on Ubiquitous meta user interfaces. 2011. ACM.
29. Luo, L., Software testing techniques. Institute for software research international Carnegie mellon university Pittsburgh, PA, 2001. 15232(1-19): p. 19.
30. Mäntylä, M.V. and K. Smolander. Gamification of Software Testing-An MLR. in Product-Focused Software Process Improvement: 17th International Conference, PROFES 2016, Trondheim, Norway, November 22-24, 2016, Proceedings 17. 2016. Springer.
31. Martin, D., J. Rooksby, M. Rouncefield, and I. Sommerville. 'Good'Organisational Reasons for'Bad'Software Testing: An Ethnographic Study of Testing in a Small Software Company. in Software Engineering, 2007. ICSE 2007. 29th International Conference on. 2007. IEEE.
32. Muntean, C.I. Raising engagement in e-learning through gamification. in Proc. 6th International Conference on Virtual Learning ICVL. 2011.
33. Myers, G.J., C. Sandler, and T. Badgett, The art of software testing. 2011: John Wiley & Sons.
34. Przybylski, A.K., C.S. Rigby, and R.M. Ryan, A motivational model of video game engagement. Review of general psychology, 2010. 14(2): p. 154.
35. Rapp, A., A. Marcengo, L. Console, and R. Simeoni. Playing in the wild: enhancing user engagement in field evaluation methods. in Proceeding of the 16th International Academic MindTrek Conference. 2012. ACM.
36. Raymer, R., Gamification: Using game mechanics to enhance elearning. eLearn, 2011. 2011(9): p. 3.
37. Rojas, J.M. and G. Fraser. Code Defenders: A Mutation Testing Game. in Software Testing, Verification and Validation Workshops (ICSTW), 2016 IEEE Ninth International Conference on. 2016. IEEE.
38. Roth, S., Serious Gamification: On the Redesign of a Popular Paradox. Games and Culture, 2017. 12(1): p. 100-111.
39. Ryan, R.M., C.S. Rigby, and A. Przybylski, The motivational pull of video games: A self-determination theory approach. Motivation and emotion, 2006. 30(4): p. 344-360.
40. Seaborn, K. and D.I. Fels, Gamification in theory and action: A survey. International Journal of Human-Computer Studies, 2015. 74: p. 14-31.

41. Shah, H. and M.J. Harrold. Studying human and social aspects of testing in a service-based software company: case study. in Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering. 2010. ACM.
42. Singer, L. and K. Schneider. It was a bit of a race: Gamification of version control. in Games and Software Engineering (GAS), 2012 2nd International Workshop on. 2012. IEEE.
43. von Ahn, L. Duolingo: learn a language for free while helping to translate the web. in Proceedings of the 2013 international conference on Intelligent user interfaces. 2013. ACM.
44. Von Ahn, L., B. Maurer, C. McMillen, D. Abraham, and M. Blum, recaptcha: Human-based character recognition via web security measures. Science, 2008. 321(5895): p. 1465-1468.
45. Whyte, E.M., J.M. Smyth, and K.S. Scherf, Designing serious game interventions for individuals with autism. Journal of autism and developmental disorders, 2015. 45(12): p. 3820-3831.
46. Wieringa, R. Design science as nested problem solving. in Proceedings of the 4th international conference on design science research in information systems and technology. 2009. ACM.
47. Witt, M., C. Scheiner, and S. Robra-Bissantz, Gamification of online idea competitions: Insights from an explorative case. Informatik schafft Communities, 2011. 192.