Gamification of Exploratory Testing Process

Savas Ozturk
savasozturk@mu.edu.tr
Mugla Sitki Kocman University, Digital Game Design Department
Mugla, Turkiye
TUBITAK BILGEM, Software Testing and Quality Evaluation Laboratory
Kocaeli, Turkiye

ABSTRACT

Exploratory testing is an experience-based testing technique often used in addition to formal ones. However, it can be used as a replacement for formal techniques when time is limited and/or documentation is poor. Conducting test charters and maximizing efficiency are seen as tough works due to their informal nature. Dividing test load to testers, scoring the severity level, consolidating the results, and reporting the issues are some challenges of this process. In this experience study, efforts of gamifying exploratory testing process are told. A public institution in Turkiye demanded software testing for their software for a limited timeframe and employed testers were motivated by gamification efforts. It was seen that gamification helped us detecting more critical faults quickly. Failure cases such as test contest organization attempts are told as well.

CCS CONCEPTS

• **Software and its engineering** → **Software verification and validation**; • **Applied computing**; • **Human-centered computing** → Empirical studies in collaborative and social computing;

KEYWORDS

exploratory testing, gamification, software testing, manual testing

ACM Reference Format:
Savas Ozturk. 2022. Gamification of Exploratory Testing Process. In *Proceedings of the 1st International Workshop on Gamification of Software Development, Verification, and Validation (Gamify '22), November 17, 2022, Singapore, Singapore.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3548771.3561411

1 INTRODUCTION

According to the ISTQB terminology, exploratory testing is mentioned among experience-based testing methods [8]. The benefits of specification-based testing such as boundary-value analysis and state transition graphs are limited, some faults can merely be detected by further investigation. Exploratory testing is conducted by expert testers who have deep experience in testing systems similar to the product under test. Their experience allows them easily catch potential drawbacks, focus on the problematic areas, and grasp faulty code. They don't spend time on regular documentation and needless routine test cases [2, 12].

On the other hand, finding the rights experts, motivating them, and paying them for ambiguous tasks are issues waiting to be solved. Functional tests are well-documented, the testing duration can be predicted and project manager knows how much tester or human power is required. Exploratory test doesn't have boundaries, it is hard to measure its efficiency. If exploratory testing is the selected test technique, testing experts should be motivated in different ways.

Gamification is a process and it is defined as employing game mechanics in a non-game application area [10]. Gamification includes some tools and techniques like score tables, badges, rewards, punishments, missions, targets, visualizations, etc. about the targeted jobs. The most reported benefits are gathered in the education domain but in recent years, companies and institutions are searching for innovative ways for increasing efficiency through gamification. Software development life cycle is one of the areas in which gamification is applied successfully [4, 5, 13].

In this study, we tell about a gamification strategy for testing a public institution software. The test process includes static code analysis, usability analysis, and exploratory testing. Gamification is mainly applied to the latter. Our test team was not an organic part of the development team, the demand for testing was very late, there was poor documentation and the management insisted on hearing a to-go or not-to-go decision in a short while. For these reasons, we preferred exploratory testing. We built up a test team including three testers and of them was also a test leader who reports to the managers weekly. Testers are employed in an unusual way, by Hungarian Assignment Algorithm for assigning tasks [9]. Each tester is given weekly targets and weekly reports are arranged by the test manager. Some bug-finding contests are conducted to make fun and to increase quality as well. Consequently, gamification efforts in exploratory testing result in customer satisfaction
In the second section exploratory testing, and in the third section gamification in the software development process is told. The fourth section lists threats to validity and mitigation precautions. In the fifth section, details about the practice of gamification in the exploratory testing process is told. In the sixth section, the benefits and failures are discussed and finally, results are presented.

Gamify '22, November 17, 2022, Singapore, Singapore

ACM ISBN 978-1-4503-9454-3/22/11…$15.00
https://doi.org/10.1145/3548771.3561411

2 EXPLORATORY TESTING AND GAMIFICATION CONCEPTS

Exploratory testing is a practical test approach applied when there is no or poor documentation and when there is limited time. Testers focus on test execution much more than planning. Tests are conducted in test charters which last only a few hours. In this period, formal test techniques such as boundary value analysis may be employed as well. Notes are taken or video&audio recording is used instead of reporting formally. In the end, generally, tests are finished with a report. This technique is an experience-based technique that supplements formal testing and according to Black et al., the most serious defects have been found by exploratory testing [7].

There are some challenges when a team of testers must be charged for exploratory testing:

- Fair task assignments for the testers may be a problem when there is limited information about the product.
- Motivating testers for re-tests or confirmation tests may be difficult due to its manual nature.
- The task of consolidating the results and reporting to the executives may be neglected and unassigned.

To overcome these negative concerns, gamification may help managers arrange the test load. Using gamification in software development and testing is not a new idea. In their recent study, Tommaso and Luca describe the incorporation of gamification mechanics in the practice of exploratory GUI testing [6]. They use leaderboards, progress bars, avatars, badges, market place, and injected bugs to make the testing process efficient. Their findings prove that gamification increases user efficiency and the most effective gamification element is the progress bar. Based on a multi-vocal literature review, the most frequently used gamification elements are points, awards, and stories [10]. According to [10], gamification results in positive effects with both test automation and exploratory testing

Stol et al. monitored an organization that installed an internal gamification platform [13]. They collected data through user surveys and a version control system. Their main objective was to find the relationship between gamification and job satisfaction. Their findings were meaningful: Gamification increases engagement, and engagement increases job satisfaction. Similar to this study, we aim to make our testers engage in the exploratory testing process which contains lots of ambiguity.

3 THREATS TO VALIDITY

The validity of our study could be threatened by its unusual nature. The People and organizations resist change. Voluntariness is the key factor of gamification. To mitigate validity threats, the test leader spent efforts like a tester to make the hierarchy invisible. Everyone in the team selected their modules to test, they only focused on finding bugs instead of living the stress of a release.

4 DESIGN AND IMPLEMENTATION OF GAMIFICATION IN EXPLORATORY TESTING

The main objective of this experience report is to tell the original application of a gamification process to a short-term testing project. A public institution requested to test their software which would be evaluated for public use. They had five developers, a project manager, and several mature modules to test. Before giving the decision to to-go, our approval was a must as an accredited test laboratory. Limited time and managerial pressure forced the test team to detect critical issues as soon as possible rather than documenting and formal test planning.

4.1 Establish The Test Team

We quickly established a test team consisting of three testers, one of them was the leader who was responsible to report to the managers of both parts. The test lead also participated in the testing process to direct and motivate the team members. The best way to follow was to test the system as much as practical instead of spending time with documentation, so the testers focused mostly on a kind of bug bounty race.

4.2 Sort The Modules According To Their Quality

To decide "where to start to test?", we sorted software modules according to their quality measurements. We used lines of code, cyclomatic complexity [11] and Chidamber and Kemerer's object-oriented design metrics suite [3] for evaluating module quality. Given that modules with lower quality tend to contain more defects [1, 11]. Before kicking off this gamification run, an improvement study on the code was held. But software developers were under pressure and they resisted making significant refactoring of the code, instead, they preferred to focus on the code parts where detected bugs are reported.

We adapted a quality scoring system from several studies [1, 3, 11]. Unit complexity was our primal priority. If the module includes methods which have more than 50 cyclomatic complexity value, they are marked as very-high complex methods. Coupling between objects (CBO) and Lack of Cohesion of Methods (LOCM) metrics from CK metrics suit helped us for classifying modules at the object level.

4.3 Task Assignments

The most important challenge was assigning modules to the testers. We asked testers to score their willingness to the modules. Before this assignment, the development team of the customer institution delivered training about the modules and their functions. We gave the testers the information about module quality as well. We had three testers, so we assigned the tasks iteratively. The testers' willingness' was based on their interests, experience, and strategy. At each iteration, modules were listed with their names, quality metric values, and the developer's name. Some testers voted for the developer (developer and tester of the modules worked tightly-coupled to complete the tests on the module), and some voted for the complexity of the module. Beyond all, it was fun at all, they liked scoring the modules and waited for the results with interest. In the first iteration, the last three modules in the list are asked to the testers sorting the modules by their quality in descending order. According to the scores, one-by-one assignments were done

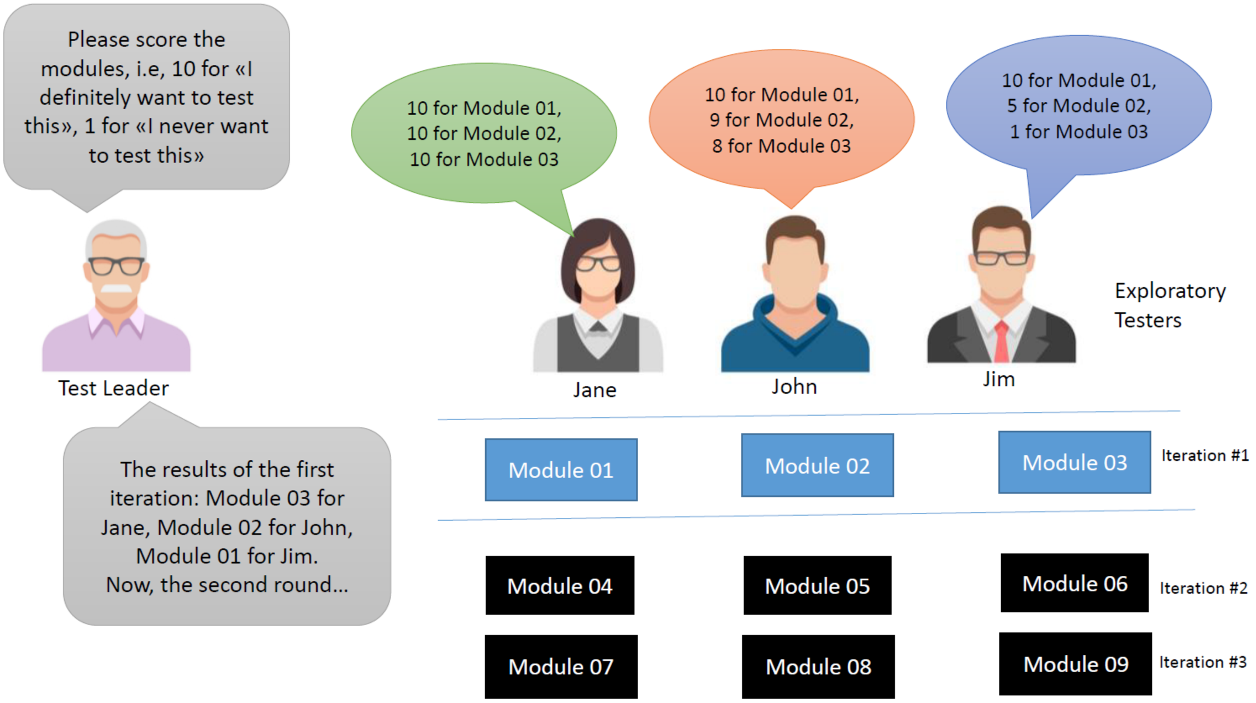

Please score the modules, i.e, 10 for «I definitely want to test this», 1 for «I never want to test this»
10 for Module 01, 10 for Module 02, 10 for Module 03
10 for Module 01, 9 for Module 02, 8 for Module 03
10 for Module 01, 5 for Module 02, 1 for Module 03
Exploratory Testers
Test Leader
Jane
John
Jim
The results of the first iteration: Module 03 for Jane, Module 02 for John, Module 01 for Jim. Now, the second round...
Module 01
Module 02
Module 03
Iteration #1
Module 04
Module 05
Module 06
Iteration #2
Module 07
Module 08
Module 09
Iteration #3

Figure 1: Using Hungarian Algorithm for Assigning Test Tasks Iteratively

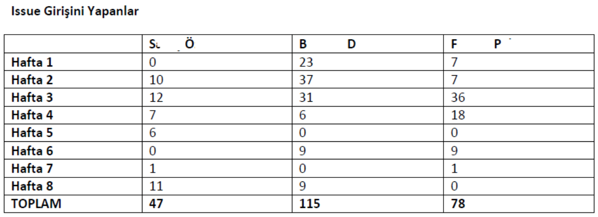
using the Hungarian algorithm. Hungarian algorithm, also known as Kuhn's algorithm, was published by Harold W. Kuhn to solve assignment problems in polynomial time [9]. The reason for using this algorithm was to make a deterministic decision for testers' choices. As expected, none of the testers made an objection to the algorithm's decision, but lately, they learned the mechanism and they started to score the tasks in a more strategically. For example, they learned that giving the highest score to all of the modules was a waste strategy. Figure 1 shows a representation of a sample task assignment process. As seen from the figure, giving the same highest score to all the modules means that "it doesn't matter, I may test whatever comes". Module 01, Module 02, and Module 03 in the example are the worst modules in terms of quality metrics such as complexity, modularity, coherence, and coupling. This assignment technique guarantees the testing of the most critical section of the software in the early stage of the process.

4.4 Weekly Sprints

Similar to the agile process, the testers were asked to test their assigned modules in weekly periods. At the end of each week, the test leader consolidated test results in terms of defect count, defect severity, and the module information with version. Also, the development team was informed to solve the defects in the shortest time. At the end of each test week, the parts met and decided to continue or stop. The testing process was repeated for eight iterations.

Weekly defect reports were like scoreboards for testers. Testers reported more defects to get more respect. The main benefit of weekly reports and score tables was to find as many bugs as possible, especially the more critical ones. Testers not only reported defects additionally they suggested 65 features or feature improvements in addition to 175 detected bugs.

Figure 2 shows a fragment from the weekly reports and it provides a list of weekly counts for reported issues.



Issue Girişini Yapanlar

	S. Ö	B D	F P
Hafta 1	0	23	7
Hafta 2	10	37	7
Hafta 3	12	31	36
Hafta 4	7	6	18
Hafta 5	6	0	0
Hafta 6	0	9	9
Hafta 7	1	0	1
Hafta 8	11	9	0
TOPLAM	47	115	78

Figure 2: Leader Board: Weekly Count of The Issues Reported by The Testers ('Hafta': 'Week', 'Toplam':'Total', 'Issue Girişini Yapanlar':'The Testers By Issue Count', names of the testers are masked)

4.5 Final Decision and Stopping The Tests

After eight iterations, both parts decided to stop the testing process based on the weekly defect table, Because the critical bugs were solved and re-tested, and the test team's dedication to this process started to become costly and redundant. The institution manager of the product sent a celebration letter to the test team because of their satisfaction with the testing quality, because no one expected such a fruitful defect-finding process at the start. The celebration letter was a great reward for the team members that it helped the team for promotion. Several projects were started and were offered depending on this successful working model. The product quality is improved because only 2 of the detected bugs in 175 bugs remained unsolved. Decision-makers approved that it was ready for public use. The lesson learned from this process was the necessity of the announcement of the great reward at the inception.

4.6 A Gamification Failure: Testing Contest

Another attempt to add some fun and find more critical bugs failed. That attempt was organizing a testing contest like a popular talent show: The Voice USA ('O Ses Turkiye', Turkish version). The name of the contest was adapted to 'O Test Turkiye' and we also prepared a poster shown in Figure 3 to attract the testers' attention. The main objectives were finding the most critical defect in a limited time and reporting the issues in detail. Rewards were symbolic, such as some stationary stuff. Jury members were experienced laboratory staff who had academic backgrounds. The competitors were from the test team and they were expected to persuade the jury about their findings are the most crucial ones for the project. This idea remained idle because of the reluctance of candidate testers: Experienced testers refused to enter such a risky challenge which would harm their esteem in front of their younger colleagues. The most important prerequisite of games and gamification is to make everyone volunteer for participating. We failed at obtaining this, and eventually, we canceled this contest.

Figure 3: The Emblem of The Contest: O Test Turkiye

5 CONCLUSION

This work summarizes gamification applied to the exploratory testing process. A public institution's software was tested and the test team accomplished a though job by satisfaction. The elements of this work were sorting the tasks according to their quality, assigning tasks to testers by optimal allocation, motivating testers by weekly scoreboards, and finally wrapping with a celebration letter from the top managers. Age difference in the team was the biggest barrier to adapt gamification. Gamification has the potential to increase efficiency and satisfaction, but the right application of gamification requires great effort and managerial support. It is not an easy game to convince all the team members to participate and this process completely depends on voluntariness. Future work will be on the comparison of the gamification effect on specification-based and experience-based test techniques.

REFERENCES

[1] Robert Baggen, José Pedro Correia, Katrin Schill, and Joost Visser. 2012. Standardized Code Quality Benchmarking for Improving Software Maintainability. *Software Quality Journal* 20, 2 (jun 2012), 287–307. https://doi.org/10.1007/s11219-011-9144-9

[2] Shuib Basri, Thangiah Murugan, and Dhanapal Durai Dominic. 2019. Validating the Conceptual Framework with Exploratory Testing. In *Software Engineering Methods in Intelligent Algorithms*, Radek Silhavy (Ed.). Springer International Publishing, Cham, 305–317.

[3] S. R. Chidamber and C. F. Kemerer. 1994. A Metrics Suite for Object Oriented Design. *IEEE Trans. Softw. Eng.* 20, 6 (jun 1994), 476–493. https://doi.org/10.1109/32.295895

[4] Gabriela Martins de Jesus, Leo Natan Paschoal, Fabiano Cutigi Ferrari, and Simone R. S. Souza. 2019. Is It Worth Using Gamification on Software Testing Education? An Experience Report. In *Proceedings of the XVIII Brazilian Symposium on Software Quality* (Fortaleza, Brazil) *(SBQS'19)*. Association for Computing Machinery, New York, NY, USA, 178–187. https://doi.org/10.1145/3364641.3364661

[5] Gordon Fraser. 2017. Gamification of Software Testing. In *2017 IEEE/ACM 12th International Workshop on Automation of Software Testing (AST)*. 2–7. https://doi.org/10.1109/AST.2017.20

[6] Tommaso Fulcini and Luca Ardito. 2022. Gamified Exploratory GUI Testing of Web Applications: a Preliminary Evaluation. In *2022 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. 215–222. https://doi.org/10.1109/ICSTW55395.2022.00045

[7] D. Graham, R. Black, and E. van Veenendaal. 2021. *Foundations of Software Testing ISTQB Certification, 4th edition*. Cengage Learning. https://books.google.com.tr/books?id=mOwxEAAAQBAJ

[8] International Software Qualifications Board (ISTQB). 2022. International Software Qualifications Board (ISTQB). https://www.istqb.org/certifications/certified-tester-foundationlevel/. [Online; accessed 13-September-2022].

[9] Harold W. Kuhn. 1955. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly* 2, 1–2 (March 1955), 83–97. https://doi.org/10.1002/nav.3800020109

[10] Mika V Mäntylä and Kari Smolander. 2016. Gamification of Software Testing - An MLR. 2 (2016), 611–614. https://doi.org/10.1007/978-3-319-49094-6

[11] Thomas McCabe. 1976. A Complexity Measure. *IEEE Transactions on Software Engineering* 2 (1976), 308–320.

[12] Christopher J. Schaefer and Hyunsook Do. 2014. Model-Based Exploratory Testing: A Controlled Experiment. In *2014 IEEE Seventh International Conference on Software Testing, Verification and Validation Workshops*. 284–293. https://doi.org/10.1109/ICSTW.2014.31

[13] Klaas-Jan Stol, Mario Schaarschmidt, and Shelly Goldblit. 2022. Gamification in Software Engineering: The Mediating Role of Developer Engagement and Job Satisfaction. *Empirical Softw. Engg.* 27, 2 (mar 2022), 34 pages. https://doi.org/10.1007/s10664-021-10062-w