Gamification of Software Testing - an MLR

Mika V. Mäntylä✉, University of Oulu <mika.mantyla@oulu.fi>

Kari Smolander, Aalto University, <kari.smolander@aalto.fi>

Abstract: This paper presents an initial multi-vocal literature review that extracts ideas for gamification of software testing. We surveyed the type of testing, system under test, role of individuals, gamification elements, challenges and drawbacks, support constructs and tools, and empirical evidence from academic sources and grey literature. Ideas were given to both automated unit-testing, and end-user related testing done by exploratory testers and beta testers. The most frequent gamification elements were points (13 sources), awards (4), stories (4), badges (3), rankings (3), levels (3) and time-pressure (3).

Keywords: Gamification, Testing, Multi-vocal literature, Grey literature

1 Introduction

Gamification is the utilization of game elements outside the context of computer games. Its purpose is to increase the engagement, motivation and performance of the participants [1]. We study gamification of software testing as software testing costs are high (35% of the IT costs [2]), testing is often an undervalued job, and testing produces lots of information in the form of numbers. The numbers can be turned to points that are a fundamental gamification element. The high cost, low appreciation, and the apparent gamifiability of testing make it an excellent target for gamification.

Our research method was a multi-vocal literature review (MLR). MLRs aim at studying all types of writings on a particular topic [3]. Thus, in comparison to traditional systematic literature review they also include grey literature such as web-pages, blog posts and discussion forum content. They are suitable for topics where academic literature is lacking due to the recent emergence of the topic or for some other reason.

When searching the literature, we performed two main steps. First, we used a recent SLR of gamification in Software Engineering [1] as a starting point for forward and backward snowballing academic literature. We studied the SLR and all primary studies it was refereeing and found three papers that were related to software testing, verification and validation. Forward snowballing additionally revealed two additional studies. Second, we used Google Search Engine to search for relevant grey literature. Our search string was "gamification software testing". We utilized the page-rank algorithm. During the process, the quality of the grey literature quickly deteriorated as we progressed further. Our final included grey literature source was 34th hit provided by Google. Overall, our goal was not to cover all possible corners but to find a reasonable amount of sources (n=20) to enable the collection of potential ideas and re-

adfa, p. 1, 2011.

quirements for a full design science project, where a software testing gamification environment is built. Our resulting spreadsheet is online [4] and sources are referred with "S" and a number of the reference, e.g. S1 refers to the first source.

2 Results

2.1 Types of Testing, Systems Under Test (SUT) and Roles of Individuals

First, we wanted to understand what types of testing gamification has been proposed for gamification. Unit testing was the most popular option mentioned in six sources (see our online spreadsheet for details). In particular, Test-driven development, a specialized way of unit testing was mentioned in three out of the six papers that mentioned unit-testing. Six sources did not mention the type of testing or talked about testing in general with no particular focus area. Two sources mentioned beta-testing as the type of testing suitable for gamification. Two mentions were also given to exploratory testing. To summarize, gamification was mostly suggested to the very opposite ends of testing. Unit-testing is technical and typical performed by developers who also develop the software. On the other hand, beta-testing and exploratory testing is often done from end-users' perspective to find out problems particularly related to the user or the customer or the domain rules were the software is used.

Second, SUT can influence whether the gamification of software testing is desirable. However, the majority of our sources (13) did not specify the system under test in detail or claimed that is applicable to all systems types. The only type of system receiving more than one mention were Games with two sources. Other systems mentioned once were: Data-center software, Inter-active system with complex workflows, Enterprise Systems, Java, and an artificial element with linked list for education purposes. Overall, it appears that any type of SUT would be suitable for gamification.

Third, as past work suggests that testing is also performed by many roles (testers, developers, customers, product managers and help-desk personnel) [5,6], we investigated the roles who participate in the gamified testing. Nine sources mentioned developers. Software testers were another notable group mentioned in eight sources. Other roles received considerably less mentions: students were mentioned three times, beta-testers and managers were mentioned in two sources and one source mentioned customer support, designers, and crowdsourced workers. Additionally, one source was ambiguous with respect to roles.

Although the literature of gamification recognizes that multiple roles can participate in gamified software testing efforts, still majority of our source focus on the two obvious groups developers and testers.

2.2 Gamification Eelements

We classified the type of gamification elements [1,7] of our sources. Gamification elements refer to constructs that try to transform work, i.e. software testing in our case, to a game. Transforming work that is supposedly boring and tedious to a game that supposedly is fun and engaging is the key motivator of gamification.

As in prior work [1] points were the most frequently mentioned gamification element with 13 sources. Points are a basic element in gamification that can be turned to other elements such as awards (4 sources), badges (3), and rankings (3). Empirical evidence of the effect of the gamification elements was limited in our sources, but for example S20 mentioned that a leaderboard (a public ranking based on points) was the most effective gamification element.

Other game elements were also frequently present. Stories or quests were mentioned in four sources. For example, S5 states that *"Testers will be impersonating different characters from a detective in industrial London…."* Time-pressure was an element mentioned in three sources. Time-pressure can increase efficiency and make games more engaging. Levels, mentioned in three sources, enable iterative progress and adjust the difficulty of the game so that a suitable challenge is always present, e.g. to complete level 1 one must have one unit-test for all classes, for level 2 one must have unit-tests for all methods and so on. Tips, mentioned in one source, can be also be used to make the games engaging and reduce deadlocks.

For an organization, games may require task distribution mechanics (mentioned in two sources). Rules, mentioned in two sources, are also important as otherwise someone can start getting points with inappropriate ways, e.g. to write meaningless code to increase unit-test coverage.

2.3 Empirical Evidence, Support Constructs, and Challenges

Only three sources provided properly reported empirical results. S12 presents two industrial case studies with impressive numbers that support the adoption of gamification. The first increased the defect fixing speed while the other focused on using static analysis results to motivate developers to create higher code quality. S13 presents a student experiment in unit-testing showing that the treatment group (gamified) found significantly more defects and had higher requirements coverage than the control group. Finally, S20 performed an industrial case study and found that their system was successful in motivating developers to write more and higher quality unit tests.

Twelve sources presented support constructs to gamification. S3 tells how their tool for gamification of testing failed and how it could be improved. S4 provides a storytelling scaffolding with roles and principles that help with gamification of testing. S7 connects gamification idea to the testers' career path. S15 provides a rule system. Actual tools were provided as JIRA add-ons (S8), Eclipse plugins (S10, S11, S13), web-based learning environments (S18), and as GUnit tool (S20).

From our sources, we found three challenges in the gamification of software testing that we consider notable. First, S6 highlighted that people have different ways to achieve the same goal. If one then starts to measure things like the quality of defect reports in the defect tracking system, then other qualities like excellent face-to-face communication that can be used as a substitute for the poor quality of written reports would be ignored. Thus, the game would reward only a selected subset of the bug reporting process, i.e. the one that is visible in the defect tracker. This ignores many relevant parts of the process as pointed out in the literature [8]. Second, even counting bugs will introduce challenges, for example counting and awarding based on the bug count could lead to a situation where five spelling mistakes in the application would result in five bug reports (S1). Such issues need to be dealt with some types of rules and game referees that decide what is right. Third, S2 pointed out a need for balance to give individuals enough freedom so that the game stays engaging and allows creativity to blossom, while still maintaining control and coordination.

3 Conclusions

We make four findings in this paper. First, gamification proposals were given to both ends of testing – automated unit-testing that is technical, and end-user related testing, i.e. beta-testing and exploratory testing. Second, the multitude of different roles and crowdsourcing in testing were recognized. Third, numerous gamification elements were present. Points were the most popular while stories appeared as something that could be used to increase the engagement in ways that are not possible for numeric point based approaches. Fourth, problems of gamification were discussed, e.g. the gamified approach might not allow employees to work in a way that is the most natural for each individual, thus, resulting in unfairness and lower productivity.

4 References

1. Pedreira O, García F, Brisaboa N, Piattini M. Gamification in software engineering–A systematic mapping. Information and Software Technology. 2015;57:157–68.

2. Buenen M, Walgude A. World Quality Report 2015–2016. Capgemini, Sogeti und HP. 2015;

3. Garousi V, Felderer M, Mäntylä MV. The need for multivocal literature reviews in software engineering: complementing systematic literature reviews with grey literature. ACM; 2016. p. 26.

4. Mäntylä MV, Smolander K. Spreadsheet of sources: Gamification of Software Testing - an MLR [Internet]. Figshare;doi: 10.6084/m9.figshare.3756600

5. Mäntylä MV, Itkonen J, Iivonen J. Who Tested My Software? Testing as an Organizationally Cross-Cutting Activity. Software Quality Journal. 2012;20:145–72.

6. Prechelt L, Schmeisky H, Zieris F. Quality experience: a grounded theory of successful agile projects without dedicated testers. ACM; 2016. p. 1017–27.

7. Hamari J, Koivisto J, Sarsa H. Does gamification work?--a literature review of empirical studies on gamification. IEEE; 2014. p. 3025–34.

8. Aranda J, Venolia G. The secret life of bugs: Going past the errors and omissions in software repositories. IEEE Computer Society; 2009. p. 298–308.