



Title	Towards a Gamified Equivalent Mutants Detection Platform
Authors(s)	Laurent, Thomas, Guillot, Laura, Toyama, Motomichi, Ventresque, Anthony, et al.
Publication date	2017-03-17
Publication information	Laurent, Thomas, Laura Guillot, Motomichi Toyama, Anthony Ventresque, and et al. "Towards a Gamified Equivalent Mutants Detection Platform," March 17, 2017. https://doi.org/10.1109/ICSTW.2017.70 .
Conference details	10th IEEE International Conference on Software Testing, Verification and Validation (ICST), Tokyo, Japan, 13-17 March 2017
Item record/more information	http://hdl.handle.net/10197/8411
Publisher's version (DOI)	10.1109/ICSTW.2017.70

Downloaded 2026-05-02 00:30:21

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)



© Some rights reserved. For more information

Poster: Towards a Gamified Equivalent Mutants Detection Platform

Thomas Laurent^{*†‡}, Laura Guillot^{*†}, Motomichi Toyama[‡], Ross Smith[§], Dan Bean[§], Anthony Ventresque^{*}

^{*}Lero@UCD, School of Computer Science, University College Dublin, Ireland Email: anthony.ventresque@ucd.ie

[†]Ecole Centrale de Nantes, France. Email: laura.guillot@eleves.ec-nantes.fr

[‡]Keio University, Yokohama, Japan. Email: {thomas.laurent, toyama}@keio.jp

[§]Microsoft Corporation, Skype Division, Seattle, USA. Email: {rosss, danbean}@microsoft.com

Abstract—This poster¹ presents a gamified system for equivalent mutants detection. This system can be used as a standalone tool for developers and testing teams alike - but we plan to use this system on a crowdsourcing platform to evaluate the various parameters involved in the detection of equivalent mutants, such as, expertise (coding and testing), familiarity with the code base, complexity of the code and tests, measured likelihood of equivalent mutants.

I. INTRODUCTION

Mutation testing is one of the most efficient and thorough test criteria available [1], [2]. It is based on seeding simple syntactic changes in programs, thus creating mutants of the programs, and measuring the capability of tests to detect these mutants. Despite its efficiency, mutation testing is plagued by a major problem: equivalent mutants [3]. An equivalent mutant is an undetectable mutant that appears when the introduced syntactic change does not result in a semantic change. These mutants artificially lower the estimated quality of tests and make mutation results noisy and hard to interpret. As automatic detection of such equivalent mutants is an undecidable problem [4], detection of these mutants is left as an additional workload to the tester.

Gamification refers to the application of techniques used in recreational activities for non playing contexts [5] - and has proven to be successful in domains such as Machine Translation [6] or Software Quality Assurance [7].

While gamified applications have been proposed for Software Testing (e.g., [8]), it is not clear whether gamification is suitable for equivalent mutants detection, as it is a skill- and labour-intensive task (experts took between 15 minutes [3] and 30 minutes [9] to label a mutant, barring the time to write test cases for each killable mutant). By tailoring the experience of each player to their skill level and expertise, and presenting them with mutants that match their profile we expect to make the task less tedious and maximise player retention. In this poster we present a prototype that we intend to use to collect information about the influence of different player and mutant attribute on the equivalent mutant detection process.

We first describe our current prototype and the architecture of the platform before describing the mechanics of the final platform. We then describe future experiments we will conduct to calibrate player mutant matching before concluding.

II. ARCHITECTURE OF OUR PLATFORM AND INTERFACE OF OUR PROTOTYPE

Our platform can be easily extended, possibly with addition of new elements, or modification of existing elements, to the modular architecture (Figure 1). The “mutation” module corresponds to a mutation employed - e.g., Mujava [10]

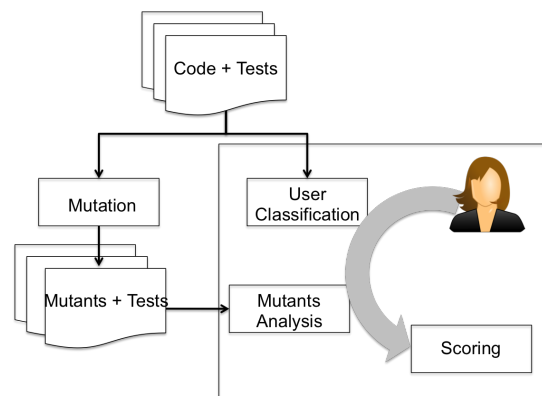


Fig. 1. System overviewte

In the current implementation² of our platform (see Figure 2), users are presented with the code (A) and the mutated code (B). Users can navigate the code base (E), write their own test to kill the mutant (C) and view some information about the mutated code (F), e.g., coverage report of the mutated class. Whenever their test seems ready they can run it against the mutated code and see the results, or they can label the mutant as equivalent (D).

III. FINAL PLATFORM

Our goal is to propose a gamified crowdsourcing tool for the labelling of equivalent mutants. As in all crowdsourcing applications, worker retention is a major concern in this project. We propose to address this concern through two main methods: gamification and mutant-worker matching. Gamification is a common way of attracting and keeping workers interested in a project while mutant-worker matching will let us present workers with mutants that match their skill level, thus avoiding boredom or discouragement. Providing workers

¹This work was supported by Science Foundation Ireland grant 13/RC/2094.

²<http://eon.db.ics.keio.ac.jp:8080/Mutation/>

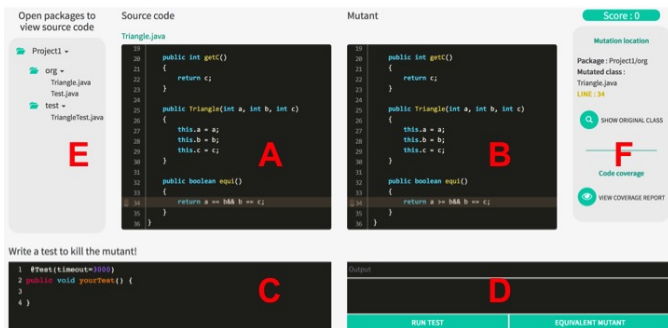


Fig. 2. Main page of our application

with challenging yet feasible tasks let them develop their skills, one of the motivations of crowd workers, as was shown in [11].

A. Gamification Mechanics

We propose a point based gamification system to bring further incentives for task completion. Users earn points every time they kill a mutant or label a mutant equivalent and enough other users agree on this labelling.

Furthermore, should a user *A* kill a mutant that was previously labelled equivalent by user *B*, *A* would receive a bonus while *B* would lose points. This prevents users from gaming the system by labelling all mutants as equivalent, thus passively gaining points, and creates a competition between users, which was shown to be effective in [12].

Further mechanics such as reward weighing based on mutant difficulty could later be introduced in order to reflect the users progress. Finally, all users can gauge their score against the others by viewing a leaderboard.

B. User Profile

To best gauge a user's ability to label mutants we build a user profile. For each user we consider the following information: the user's familiarity with the code under test, the user's qualifications, the user's history on the system...

Through this profile we want to determine each user's proficiency at labelling mutants and their expertise, on a certain piece of code or a certain type of mutants for example.

C. Mutant Profile

To estimate the complexity of labelling a mutant we build a mutant profile that includes information on the code in which the mutant is located and on the mutant itself. In this profile we consider information such as the length and complexity of the mutant's method, as defined in [13], or the impact of a mutant such as defined in [3]. We hypothesise that mutants located in complex parts of the code are harder to label, as are those with low impact as their potential impact is harder to detect.

This mutant profile lets us assess the difficulty of labelling each mutant and which area of expertise each mutant requires to be labelled.

IV. FUTURE WORK

In order to correctly match user and mutant profiles we need to evaluate the influence of each parameter on both the difficulty of a mutant to be labelled and the ability of a user to label mutants. To perform such evaluation and assess the capability of the crowd we will run experiments through Amazon Mechanical Turk³ (AMT).

To evaluate the impact of the mutant profile we will ask the crowd to label mutants of different impacts and in code bases of different length and complexity and observe the effect of these parameters on both the quality (in terms of precision compared to a ground truth) of the results obtained and the latency of the tasks. Similar experiments will be designed to evaluate the impact of the user profile parameters.

Contrary to most task distributed through AMT, labelling mutants is a complex task that requires particular skills. To accommodate the special nature of our task we will take particular care in the task design. One of the main pitfalls of open call crowdsourcing platforms is spam from malicious users. To avoid such noise in our results we will offer a reward that corresponds to the task's difficulty and will make use of AMT's qualification mechanisms, thus ensuring the capability of our workers. Such approaches have led to better results in previous studies. For example in [8], where AMT's crowd was asked to assess test oracles against software specifications, the crowd vetted through the qualification mechanism performed significantly better than the non vetted crowd.

V. CONCLUSION

In this poster we presented our prototype for a gamified equivalent mutant detection platform. Our system lets players study mutants and kill them by writing a test or label them as equivalent. Players earn points when they kill a mutant or when they agree on the equivalence of a mutant.

We will study the effects of different mutant and player attributes to best tailor the game to the players and propose the best experience possible and thus the best results for the tester.

REFERENCES

- [1] J. H. Andrews, L. C. Briand, and Y. Labiche, "Is mutation an appropriate tool for testing experiments?" in *ICSE*, 2005, pp. 402–411.
- [2] N. Li, U. Praphamontripong, and J. Offutt, "An experimental comparison of four unit test criteria: Mutation, edge-pair, all-uses and prime path coverage." in *ICST*, 2009, pp. 220–229.
- [3] B. J. Grün, D. Schuler, and A. Zeller, "The impact of equivalent mutants," in *ICST*, 2009, pp. 192–199.
- [4] T. A. Budd and D. Angluin, "Two notions of correctness and their relation to testing," *Acta Informatica*, vol. 18, no. 1, pp. 31–45, 1982.
- [5] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: defining gamification," in *MindTrek*, 2011, pp. 9–15.
- [6] L. Guillot, Q. Bragard, R. Smith, D. Bean, and A. Ventresque, "Towards a gamified system to improve translation for online meetings," in *GamifIR*, 2016, pp. 21–28.
- [7] R. Smith and L. A. Kilty, "Crowdsourcing and gamification of enterprise meeting software quality," in *UCC*, 2014, pp. 611–613.
- [8] F. Pastore, L. Mariani, and G. Fraser, "Crowdoracles: Can the crowd solve the oracle problem?" in *ICST*, 2013, pp. 342–351.

³www.mturk.com

- [9] D. Schuler, V. Dallmeier, and A. Zeller, "Efficient mutation testing by checking invariant violations," in *ISSTA*, 2009, pp. 69–80.
- [10] Y.-S. Ma, J. Offutt, and Y. R. Kwon, "Mujava: An automated class mutation system," *Software Testing, Verification and Reliability*, vol. 15, no. 2, pp. 97–133, 2005.
- [11] N. Kaufmann, T. Schulze, and D. Veit, "More than fun and money. worker motivation in crowdsourcing-a study on mechanical turk." in *AMCIS*, vol. 11, 2011, pp. 1–11.
- [12] J. M. Rojas and G. Fraser, "Code defenders: A mutation testing game," in *The 11th International Workshop on Mutation Analysis. IEEE*, 2015.
- [13] T. J. McCabe, "A complexity measure," *TSE*, no. 4, pp. 308–320, 1976.