

# Crowdtesting Intermediary Tool for Managing Public Service Software Project

Mohd Syahmi Mustapha<sup>1</sup>, Mar Yah Said<sup>2</sup>

<sup>1,2</sup>Faculty of Computer Science and Information Technology  
Universiti Putra Malaysia  
43400, Serdang, Selangor, Malaysia  
syahmi.osu@gmail.com<sup>1</sup>, maryah@upm.edu.my<sup>2</sup>

## Article Info

Volume 81

Page Number: 3201- 3206

Publication Issue:

November-December 2019

## Abstract:

Software testing is important to ensure correctness of the software, gaining confidence from stakeholders, and contributing towards achieving high quality software. One approach to conduct software testing is through crowdtesting. It allows people from the crowd to test a particular software using their own devices in real environment. Currently in public service sector there is no existing intermediary tool to manage crowdtesting activities for public service software project. Therefore, public service software project relied on common testing approaches such as testing by internal employees or outsourced to specific suppliers, that in turn making public service software projects facing the risk of inadequate testing. This study intends to determine whether the implementation of crowdtesting is able to address the problems of inadequate testing in public service software project and to propose an application as intermediary tool for crowdtesting in public service. This study employed interviews and survey with IT practitioners in public service sector to understand the applicability of crowdtesting in public service and specifications for the proposed intermediary tool. The intermediary tool is developed and evaluated to determine its effectiveness in managing crowdtesting for public service software project. The evaluation shows that most of the participants agree that the intermediary tool shows effectiveness in terms of defect detection, cost benefit, time, and testing coverage.

## Article History

Article Received: 5 March 2019

Revised: 18 May 2019

Accepted: 24 September 2019

Publication: 14 December 2019

Keywords—Software crowdtesting, crowdtesting intermediary tool, crowdtesting for public service.

## 1. INTRODUCTION

Software testing is an important phase in software development life cycle. Graham et al.(2008) highlighted that the objective of software testing is to ensure correctness of the software, finding defects as early as possible, gaining confidence from stakeholders, and contributing towards achieving high quality software. Software testing are commonly conducted by internal employees of companies or outsourced to specific suppliers as suggested by Zogaj et al. (2014) and Yan et al. (2014). Another approach

to conduct software testing is through crowdtesting that leverage on the concept of crowdsourcing by opening the testing activities for participation to the mass of people in public. Leicht et al. (2016), Alyahya and Alrugebh (2017), Hussain, A., et al. (2019), and Leicht (2018) identified crowdtesting as an approach of conducting software testing by allowing people from the crowd located in different places to test particular software using their own devices in real environment. Guaiani and Muccini (2015), and Leicht (2018) proposed that through crowdtesting project owners of the software essentially outsource the

testing activity to the mass of people and granting access for the crowd to test their software. Leicht et al. (2017) suggested that crowdtesting is more focused on the output of the software under test based on certain given input since source code usually is not available to the crowd testers. Crowdtesting provides several advantageous for software testing. Schneider and Cheung (2013), and Liu et al. (2012) argued that for testing that requires high number of testers, crowdtesting provides the avenue to attract high number of participants in relatively short time and low cost. Guaiani and Muccini (2015), and Mao et al. (2017) proposed that through crowdtesting, software can be tested in the real environment, conducted by the real user, getting quick responses from the testers, and expediting testing activities by leveraging on distributed resources in the crowd. Mao et al. (2017) also proposed that crowdtesting able to reduce the cost of internal staffing since it provides avenue for companies to engage with testers from the external crowd whenever it is required. Public service software project usually relied on common testing approaches such as testing by internal employees or outsourced to specific suppliers. Crowdtesting approach is not utilized because currently in public service sector there is no existing intermediary tool that can manage crowdtesting activities specifically for public service software project. Unlike commercial software project, public service software project requires emphasis on the identity of testers as well as confidentiality of government data and resources. Existing commercial intermediary tools do not provide adequate measures for identity of testers and confidentiality for government data and resources. Therefore, public service software projects facing the risk of inadequate testing especially for software that is designed for use of public and with multiple devices or platforms. Leicht et al. (2017) argued that to conduct testing with all the possible combination of available platforms and devices by the common testing approaches are not practical and economically not viable. Guaiani and Muccini (2015) suggested that the common testing approaches might not be suitable to test software that can run on multiple hardware such as mobile applications due to challenges that come with variety of possible configurations that will affect the behavior of the software. Baharom et al. (2016) conducted a study on current practices in software development in Malaysia found out that 73.2% of respondents viewed that current practices of software testing alone is inadequate to ensure the software under testing has

achieved its specification. The study main contribution is that crowdtesting using tool, support findings from existing literatures. Based on the result of the evaluation, conducting crowdtesting using tool open more possibilities for more defects to be detected, produces cost benefit, enable testing to be conducted in relatively short time, give more time for testing, and provide more testing coverage in terms of number of testers and devices/platforms. The remainder of this paper is organized as follows. Section 2 discusses on the existing materials related to the area of study and methods that were employed in the course of this study. Section 3 presents the results of this study and discussion on the results. Section 4 discusses on the conclusion of this study.

## 2. MATERIALS AND METHODS

### 2.1. LITERATURE REVIEW

Crowdtesting leverages on the model of crowdsourcing where the testing activities are conducted by the crowd. Alyahya and Alrugebh (2017) identified three components of crowdtesting which are crowd seeker, intermediation tool, and crowd tester. Crowd seeker is the owner of the software to be tested, crowd tester is the people in the crowd who are going to conduct the testing, and intermediation tool which responsible to link the crowd seeker with crowd tester and manage the crowdtesting processes. From this point forward, the terms crowd seeker, crowd tester, and intermediation tool will be used to represent the three components in crowdtesting. Crowdtesting can be conducted through the crowd seeker's internal tool. In this scenario, the crowd seeker will directly interact with the crowd testers and manage the crowdtesting processes. Zogaj et al. (2014) argued that in most cases crowdtesting are conducted through external intermediary tool. Intermediary tool is a component in crowdtesting that enables project owners to connect with testers from the crowd via middle party. Alyahya and Alrugebh (2017) highlighted that intermediary tool allows the project owners to specify their testing requirements, and provides access for testers in crowd to join the testing activity. Intermediary tool between the crowd seeker and the crowd testers has a huge role in crowdtesting implementation in order to manage the whole processes and activities. Zogaj et al. (2014) suggested that intermediary tool will deal with the crowd seeker to have the appropriate requirements and

design of the test as well as to communicate and manage the crowd testers in the course of the testing.

Alyahya and Alrugebh (2017) identified three main components of crowdtesting which are crowd seeker, intermediary tool, and crowd tester. Crowd seeker is the owner of the software to be tested, crowd tester is the people in the crowd who are going to conduct the testing, and intermediary tool which responsible to link the crowd seeker with crowd tester and manage the crowdtesting processes. From this point forward, the terms crowd seeker, crowd tester, and intermediary tool will be used to represent the three components in crowdtesting. Zogaj et al. (2014) suggested that crowdtesting can be conducted through the crowd seeker's internal tool. In this scenario, the crowd seeker will directly interact with the crowd testers and manage the crowdtesting processes. However, in most cases crowdtesting are conducted through external intermediary tool. Alyahya & Alrugebh (2017) highlighted that intermediary tool

allows the project owners to specify their testing requirements, and provides access for testers in crowd to join the testing activity. Zogaj et al. (2014) also suggested that intermediary tool will deal with the crowd seeker to have the appropriate requirements and design of the test as well as to communicate and manage the crowd testers in the course of the testing. Zogaj et al. (2014) identified two main approaches to conduct crowdtesting in accordance to crowdsourcing model as illustrated in Figure 1. The first approach is crowdtesting without mediation approach where the crowd seeker uses their own internal tool to conduct crowdtesting and engage directly with crowd testers. The second approach involves mediated crowdtesting where crowd seeker will engage external intermediary tool that provide crowdtesting intermediary tool and manages crowd testers. Crowd seeker will only interact with intermediary tool, whereas intermediary tool will communicate and manage crowd testers as well as the crowdtesting activities.

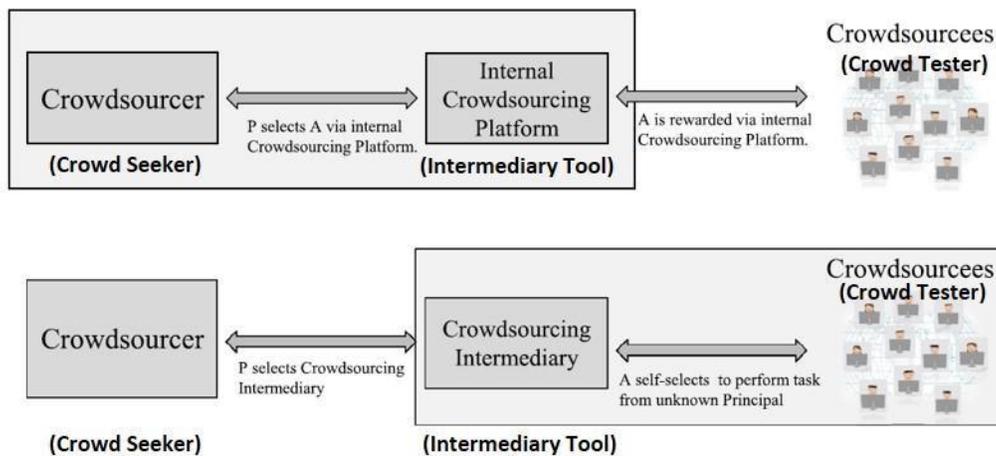


Figure 1. Approaches of crowdtesting based on crowdsourcing model (Zogaj et al. 2014).

## 2.2. INTERVIEW AND SURVEY

This study employed interview to collect the data on the applicability of the study in public service as well as to gather requirement for the proposed intermediary tool from subject matter expert in software testing and IT practitioners in public service sector. Interviews were conducted with nine respondents that consisted of eight IT personnel from public sector and one IT expert from the industry. A survey has been conducted to confirm the problem statement as highlighted in Section 1 in this article. The survey questionnaire was distributed to IT staff and government officers in public sector from which 33 responses were recorded.

All the respondents of the survey consisted of IT personnel and stakeholders in public sector that have been involved in software development project.

Result from the interviews suggested that current testing approaches practiced by public sector are inadequate to sufficiently test software with multiple combination of devices and platforms. 89% of the respondents agreed that in their previous projects, testing was conducted with limited combination of devices and platforms due to resources constraint. Results from the survey shows that on the current testing approach practiced by their organizations, 54.5% of the respondents feel that it is not sufficient

to test with multiple devices and platforms available in the public. With limited devices and platforms, the project team might not be able to test the software with devices and platforms that the end-user use. Some compatibility issues or errors of the software might not be discovered during testing and eventually, the issues and errors will only be discovered by end-users after deployment. Analysis conducted on the result from the interview and survey show that existing testing approaches practiced by public service sector usually conducted with limited combination of devices and platforms as well as only involved small number of testers. This can be challenging to comprehensively cover all test scenarios and may lead toward insufficient testing.

### 2.3. TOOL AND EVALUATION

An application was developed to implement the crowdtesting intermediary tool. The design of the intermediary tool was based on the requirements gathered through interviews with selected respondents. Based on the requirements gathered, core functionalities of the intermediary tool consist of creation of crowdtesting project by crowd seeker, selection of suitable crowd tester, reporting of test result by crowd tester, and presenting test result to the crowd seeker. The intermediary tool was developed using language PHP, Laravel PHP Framework and MySQL as the database. Evaluation was conducted to

determine the effectiveness of the proposed tool as crowdtesting intermediary tool for public service software project. The evaluation was conducted via case study on three software projects within public service sector. A total of 12 participants were involved in the evaluation that consist of one project owner and three crowd testers for each of the software project. The effectiveness of the intermediary tool was measured in term of its defect detection capability that relates to the number of defects found during the testing, cost benefit produced when conducting the crowdtesting, time required to conduct the testing that relates to the time needed to setup the testing, time needed to execute the testing, and testing time that can be covered through crowdtesting, and testing coverage provided by crowdtesting in terms of number of testers involved and devices/platforms used. The results from the evaluation were to measure the effectiveness of the intermediary tool to manage crowdtesting activities for public service software project. The evaluation was conducted through questionnaire that captured the responses from participants on the effectiveness of the intermediary tool. The questions consist of Likert scale with five levels rating scale. Participants were given the options to choose either, Strongly Disagree, Disagree, Neutral, Agree, or Strongly Agree. Table 1 shows the result of the evaluation on the effectiveness of the intermediary tool with the relevant rating scale.

Table 1. Result of the evaluation on the effectiveness of the intermediary tool

		<b>stronglyagree</b>	<b>agree</b>	<b>neutral</b>
1	<b>The intermediary tool enables more defects to be detected by testing with multiple combinations of devices or platforms.</b>	66.7%	25%	8.3%
2	<b>The intermediary tool produced cost benefit when conducting testing with multiple combination of devices or platforms.</b>	50%	50%	-
3	<b>Short time was needed to setup testing through the intermediary tool.</b>	50%	50%	-
4	<b>Short time was needed to execute testing through the intermediary tool.</b>	50%	41.7%	8.3%
5	<b>The intermediary tool enable more time to conduct testing outside office hours.</b>	75%	25%	-
6	<b>The intermediary tool enables adequate number of testers to be involved by allowing the crowd to join the testing.</b>	75%	25%	-
7	<b>The intermediary tool enables adequate testing coverage by conducting testing with multiple combinations of devices or platforms.</b>	50%	50%	-

One notable result from the evaluation is the ability of the intermediary tool to enable more defects to be detected by testing with multiple combination of devices or platforms. Through the intermediary tool, software projects can be tested with more devices and platforms other than the project owners currently possess. Software might behaves differently with different devices or platforms. Thus by testing a software with multiple devices or platforms, it will open up possibility to discover more defects as compared to testing with limited number of devices or platforms. This is essential especially for applications that are intended for use of the public or involved many types of users. The public or users might access the applications with multiple devices or platforms, thus the applications need to function correctly for all the devices or platforms involved.

### 3. RESULTS AND DISCUSSION

Through the intermediary tool, software projects can be tested with more devices and platforms other than the project owners currently possess. Testing a software with multiple devices or platforms give the possibility to discover more defects as compared to testing with limited number of devices or platforms. Furthermore, by opening the testing to the crowd through the intermediary tool, testing will be conducted by crowd testers using their own devices and platforms. Since more time is available to conduct testing since testing still can be conducted by crowd testers outside office hours such as at night and during weekends, this will allow testers ample time to thoroughly test the software. By opening the testing to be conducted by the crowd, the intermediary tool will enable more testing coverage in terms of number of testers involved in the testing. Overall, by having higher number of testers and devices or platforms involved in the testing, the intermediary tool can provide adequate testing coverage. Currently, there are multiple crowdtesting intermediary tools that are openly available for the industry and testers in the public. Zanatta et al. (2016) highlighted that some of the crowdtesting intermediary tool such as uTest, Passbrains, BugFinders, and Testbirds implement selection process through recruitment while 99tests implement selection process through competition. Yan et al. (2015) identified that most of the existing crowdtesting intermediary tools such as uTest, Pay4Bugs, Mob4Hire, Feedback Army, and 99test require manual submission of test results whereas for

crowdtesting tool iTest, test results are automatically submitted. Alyahya and Alrugebh (2017) identified the type of testing that can be conducted using some of the existing crowdtesting intermediary tool which include functional, security, usability, load, localization, automation, and compatibility testing. On the other hand, the proposed tool has been evaluated mainly for functional testing during the stage of user acceptance testing. Therefore for other type of testings, further evaluation would be required to determine the suitability of the tool.

### 4. CONCLUSION

Result from this study shows that current testing approaches practiced by public sector is not sufficient to comprehensively test software with multiple combination of devices and platforms. Furthermore, current approaches usually can only involve small number of tester since the testers need to be available for a specified time and place to conduct the test. Testing software through crowdtesting approach can address these problems by opening the test to the crowd. Through crowdtesting, testing with multiple combination of devices and platforms is possible with the use of devices and platforms available in the crowd. Furthermore, higher number of testers can be achieved since testing can be conducted outside office hours such as at night or during weekend. Through this study, an intermediary tool to manage implementation of crowdtesting in public service project was developed. The intermediary tool will enable public service software project to take benefit on the advantages of crowdtesting. Through the application of the intermediary tool, public service software projects can be tested by wide number of people and multiple combination of hardware and platforms. This can provide extensive testing coverage and increasing level of test adequacy, as well as providing opportunity to get important feedback on the usability by the crowd.

### 5. ACKNOWLEDGMENT

The authors would like to express appreciation for the support of the sponsor Public Service Department and the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia.

## 6. REFERENCES

- [1] Alyahya, S., & Alrugebh, D. 2017. Process improvements for crowdsourced software testing. *International Journal of Advanced Computer Science And Applications*, 8(6), 32-40.
- [2] Baharom, F., Deraman, A., & Hamdan, A. R. 2016. A survey on the current practices of software development process in Malaysia. *Journal of ICT*, 4, 57-76.
- [3] Graham, D., Van Veenendaal, E., & Evans, I. 2008. *Foundations of software testing: ISTQB certification*. Cengage Learning EMEA.
- [4] Guaiani, F., & Muccini, H. 2015. Crowd and laboratory testing can they co-exist?: An exploratory study. In *Proceedings of the Second International Workshop on CrowdSourcing in Software Engineering*, 32-37.
- [5] Hussain, A., Mkpojiogu, E.O.C., Kutar, M., 2019. The impact of software features' perceived importance on the perceived performance of software products' quality elements. *Journal of Computational and Theoretical Nanoscience*, 16 (5-6), pp. 2135-2140.
- [6] Leicht, N. 2018. Given enough eyeballs, all bugs are shallow—a literature review for the use of crowdsourcing in software testing. In *Proceedings of Hawaii International Conference on System Sciences (HICSS 2018)*.
- [7] Leicht, N., Blohm, I., & Leimeister, J. M. 2016. How to systematically conduct crowdsourced software testing? Insights from an action research project. In *Proceedings of International Conference on Information Systems (ICIS 2016)*.
- [8] Leicht, N., Blohm, I., & Leimeister, J. M. 2017. Leveraging the power of the crowd for software testing. *IEEE Software*, 34(2), 62-69.
- [9] Liu, D., Bias, R. G., Lease, M., & Kuipers, R. 2012. Crowdsourcing for usability testing. *proceedings of the association for information science and technology*, 49(1), 1-10.
- [10] Mao, K., Capra, L., Harman, M., & Jia, Y. 2017. A survey of the use of crowdsourcing in software engineering. *Journal of Systems and Software*, 126, 57-84.
- [11] Schneider, C., & Cheung, T. 2013. The power of the crowd: Performing usability testing using an on-demand workforce. In *Information Systems Development* (pp. 551-560). Springer, New York, NY.
- [12] Yan, M., Sun, H., & Liu, X. 2014. iTest: Testing software with mobile crowdsourcing. In *Proceedings of the 1st International Workshop on Crowd-based Software Development Methods and Technologies* (pp. 19-24). ACM.
- [13] Yan, M., Sun, H., & Liu, X. 2015. Efficient testing of web services with mobile crowdsourcing. In *Proceedings of the 7th Asia-Pacific Symposium on Internetware* (pp. 157-165). ACM.
- [14] Zanatta, A. L., Machado, L. S., Pereira, G. B., Prikladnicki, R., & Carmel, E. 2016. Software crowdsourcing platforms. *IEEE Software*, 33(6), 112-116.
- [15] Zogaj, S., Bretschneider, U., & Leimeister, J. M. 2014. Managing crowdsourced software testing: A case study based insight on the challenges of a crowdsourcing intermediary. *Journal of Business Economics*, 84(3), 375-405.