Crowdsourced Software Testing: A Timely Opportunity

Praveen Kumar Donepudi

Enterprise Architect, Information Technology, UST-Global, Inc., Ohio, **USA**

*Corresponding Contact:
Email: praveen.donepudi@ust-global.com

ABSTRACT

The concept of crowdsourcing has gained a lot of attention lately. Many companies are making use of this concept for value creation, as well as the performance of varied tasks. Despite its wide application, little is known about crowdsourcing, especially when it comes to crowdsourced software testing. This paper explores the crowdsourced software testing concept from a wider perspective ranging from a cost-benefit analysis, crowdsourcing intermediaries, and the level of expertise in the crowd. Drawing from a varied range of sources, a systematic literature review is done, where the research narrows down to ten most relevant peer-reviewed sources of high impact rating. In a comparative analysis between crowdsourced software testing and in-house testing, it is found that crowd testing has numerous advantages when it comes to efficiency, user heterogeneity, and cost-effectiveness. The study indicates that intermediaries play a key role in managing the connection between the crowd and crowdsourcing companies despite various challenges. A comparison between novice testers and expert testers reveals that both the two have their unique capabilities in their respective domains.

Key words:

Crowdsourcing, Crowdsourced Software Testing, intermediaries, expertise

INTRODUCTION

Crowdsourced software testing, simply referred to as crowd testing, is the application of crowdsourcing techniques in the testing phase of the software development lifecycle (Leicht, Blohm, & Leimeister, 2017). In this domain, software testing activities are outsourced to the crowd, who are then referred to as the crowdsources. The sourcing company can either interact directly with the crowd or use intermediary channels that can serve the company at some cost. The outsourced tasks, as well as the contracted crowds, can be very diverse depending on the kind of testing to be done, such as security testing, usability testing, and functional testing.

METHODOLOGY

To better understand the idea of crowdsourced software testing, systematic literature was done. Systematic literature is a structured procedural mechanism to identify, analyze, and interpret existing literature that is in line with the research at hand. This research method not only aggregates relevant studies but also supports the advancement of evidence-based recommendations for interested parties. The literature review procedure followed a well-structured documentation in four distinct steps, namely, planning, article searching, screening of different papers, and finally, extraction of findings.

The planning stage entailed two steps as:

- stating the research interest in the form of a research objective
- deriving a relevant search strategy

RESEARCH OBJECTIVE

To assess the value of crowdsourced software testing and determine the challenges, techniques, and level of expertise needed in the crowd.

RESEARCH STRATEGY

The search strategy entailed the identification of relevant concepts, selection of sources, and determination of inclusion and exclusion criteria.

Inclusion and exclusion criteria: A search through peer-reviewed articles since 2010 was done. Any article published before the year was excluded to enable draw latest insights with regards to software testing. In order to draw from a wide range of scholarly evidence, unpublished works such as website articles and non-scientific resources were never considered. Since the terminologies employed in this paper are English, non-English articles were excluded. Finally, the study included only full peer-reviewed scholarly articles that could be freely accessed through online databases. The articles were of high impact rating.

Search resources: All the research databases that contained articles fitting the inclusion criteria were selected. Some of these databases include; ACM Digital Library, IEEE, Wiley, Springer Link, and Ebscohost.

Identification of concepts: Based on the research question, 'Crowdsource' was the main term identified. In a way to support the choice of the search terminologies, other queries were conducted. To begin with, the phrase 'Software Testing' was applied across all databases, which brought the total number of publications to 1335. We then narrowed to the main concept under review and identified the phrase "Crowdsourced Software testing," of which the results are as shown below:

Database	Terminology/Phrase	
	Software Testing	**Crowdsourced Software Testing**
ACM	943	157
IEEE	77	36
Wiley	137	121
Springer Link	123	98
EBSCOhost	55	36
Total	**1335**	**448**

The set of articles was then limited using the same phrase taking into account only the three keywords in the phrase. The number of publications came to 53, giving a reasonable amount of articles to be evaluated within a reasonable amount of time. A careful review of the resulting papers by reading through the abstracts brought the final results to 10, which was appropriate to address the research objective stated.

DISCUSSION OF THE RESEARCH FINDINGS

Advantages and Disadvantages

Recent case studies reveal the main drivers and advantages of crowdsourced software testing in comparison to in-house testing and development. In the first place, the relative advantage of this process is largely affected by the test object (Leicht, Knop, Blohm, Müller-Bloch & Leimeister, 2016). For instance, when testing less complex software such as mobile applications, it is almost impossible for test managers to test the application's compatibility that has to be deployed to the varying platforms of the public audience. Crowdsourcing would easily address the compatibility concern hence creating relief in that case. Besides, it is easy for a crowdsourcing company to address individual target groups such as bank customers and people from different countries through the crowdsourcing technique (Leicht et al., 2016). Crowdsourced software testing also comes with time-related benefits since the process can be done quickly due to reduced internal workload peaks where the crowd can be mobilized to work any time of the day (Stol & Fitzgerald, 2014). This enhances real-time access for test managers through synchronized access to bug validation and synchronization. Crowdtesting companies can further benefit from the reliable usability insights that may otherwise have been regarded as less severe due to in-house organizational assumptions.

On the other hand, it is right to posit that crowdsourced software testing does also has disadvantages. This comes in handy, especially when the test object requires a lot of functional knowledge to conduct very specific tests such as security. There are also cases where the security of data can be of greater concern. Results submitted by crowdsourcees should undergo regular inspection since individuals have varying abilities (Liu et al., 2012). Generally, the effectiveness of the testing feedback depends on the level of security standards such that the higher the standards such as the two-factor authentication, the lesser the effectiveness. Further, extra costs would create more relative disadvantages, especially in small firms where internal communication between the testing manager and the testing department is fast and easier.

Intermediaries

In many common instances, a crowdsourcer would develop a platform and host it within internal organs. However, cheaper alternatives would entail carrying out outsourcing activities through a third-party agent, otherwise referred to as the crowdsourcing intermediary. As the name implies, crowdsourcing intermediaries provide a neutral platform where a crowdsourcees and the crowdsourcer can freely interact and set their terms of services. For this reason, crowdsourcing intermediaries play a very significant role when it comes to outsourced software testing. Crowdsourcing intermediaries are simply web-platforms or online marketplaces that create the relationship between the crowd and crowdsourcing companies (Zogaj, Bretschneider, & Leimeister, 2014). On one hand, the intermediaries query the crowdsourcing company when it comes to tasks and solution

requirements in a way that the crowd can solve the tasks correctly. On the other hand, these intermediaries manage the crowd, so the tasks are performed as per the requirements. Many studies assign a lot of importance to intermediaries in the sense that they enable the crowdsourcing firm leverage on a vast pool of human capital. For instance, Billington and Davidson (2013) posit that intermediary networks serve as valuable brokers who connect different parties, i.e., knowledge suppliers and knowledge seekers.

There are three main challenges associated with crowdsourcing intermediaries, namely; process management, crowd management, and technology management. Process management involves treatments and procedures implemented by an intermediary while managing all the software testing activities. In line with Muhdi and Boutellier's (2011) findings on idea generation process, the first challenge when it comes to crowdsourced software testing is the appropriate definition of the testing requirements, which works to ensure that testing by any undefined crowd heads in the right direction. Most articles on crowdsourcing recommend breaking down a complex task into smaller, manageable tasks. Mechanisms to implementing quality assurance have to be incorporated within a crowdsourcing process, failure to which there would be a likely risk of incurring valueless outcomes out of a crowdsourcing process (Daniel et al., 2018).

Effect of expertise

Depending on the specific needs of firms involved in crowdsourced software testing, characteristics and roles of contributors may vary substantially. Some software testing activities might entail the processing of less complex tasks in terms of structural complexity and effort intensity. This way, the value of the crowdsourcing process would stem from the efficiency and least costly measures of completing large scale tasks. However, the value of the process might revolve around intellectually complex tasks that may need a lot of expertise within a specific field. These distinct types of software testing tasks would thus affect the crowd differently. Some crowds would consist of generally any individual, some would need people with a particular profile, and others would demand people with expert knowledge, skills, and unique capabilities. It is thus crucial that crowdsourcing initiatives be aligned to workers based on some suitability criteria. The value of the tested outcome is said to be functionally dependent upon the problem to be solved, the complexity of the process, and the degree of crowd-workers' expertise (Leicht, Rhyn, & Hansbauer, 2016). However, it is still not clear when it comes to different task implications and configurations of the crowd worker with regard to performance.

Depending on the level of complexity, crowdsourced tasks create different demands when it comes to the expertise or characteristics of workers. In this regard, one of the primary mechanisms that determine testing outcomes is human expertise. Leicht, Rhyn, and Hansbauer (2016) define expertise as outcomes emanating from the acquisition of a large amount of knowledge or systematic skill in a particular discipline. The differences emanating from the level of expertise have a great influence on human performance and, therefore, the testing outcomes (Leicht, Rhyn, & Hansbauer, 2016). On the flipside, novices and expert workers differ in how their knowledge level is manifested while performing a task. Even though both the novice and expertise use the same descriptors for a task, the cues from experts utilize a high level of tacit knowledge compared to the novice. The problem-solving techniques for expert workers employ a high level of procedural knowledge characterized by outright applicability conditions. To the contrary, the novice lacks abstract methodologies even though they may possess the declarative knowledge regarding a task.

These observations, as can be drawn from this research, are likewise applicable to computer science and, in particular, software testing.

However, previous study (Donepudi, 2017) indicate that experts do not necessarily outperform the novices in all domains relevant to expertise. For instance, novices will outperform experts when it comes to tasks that require the memorization of a superficial outlook of a task. Yes, while experts are good at understanding the deeper structures of functionality, they have a tendency to overlook surface features and, therefore, finer details (Foong, Gergle, & Gerber, 2017). In this regard, expert people would do better in unraveling abstract concepts, but novices would outperform them when unraveling concrete issues related to certain software. Furthermore, expert workers are likely to be functionally fixed and mentally biased when evaluating software. When a task is running contrary to the normal procedures, novices would be of great help as opposed to experts who may inhibit creative problem solutions in many circumstances.

IMPLICATIONS

This study makes a very valuable contribution to the crowdsourced software testing literature. In the first place, the overall concept presents the techniques associated with crowdsourcing and its relative advantage. This provides a strong foundation to explain why software companies would need to crowdsource rather than over-rely on internal mechanisms for problem-solving. Also, it is our intention that this research would be used to explore the concrete crowdsourcing mechanisms not only in software testing but also in other tasks of valuable interest. In illustrating the relative advantages associated with crowdsourced software testing, the research helps to offer insights on how crowdsources can gain a competitive advantage through the process. This research would also contribute to the existing literature when it comes to crowdsourcing expertise in two ways. Firstly, the research provides valuable insights into the process of design and resource allocation to crowd workers in ways that would increase the performance of crowd workers. When it comes to the particular domain of software testing, this research aims to support the decision-making process involved in crowdtesting configurations. The findings may be valuable in helping decide which forms of testing to be crowdsourced and which ones should be tested using alternative approaches.

CONCLUSION

The concept of crowdsourced software testing is still in its developmental stages. This is not only evidenced by the dates of publications in the existing literature but also the fact that existing research is well dominated by case studies and experimentations in different scenarios. From the experimentation and literary perspective, the concept seems to present valuable solutions in terms of quality, cost-effectiveness, and efficiency. However, limited adoption of the concept in the real-world scenarios is still evident. Therefore, in order to explore the topic and draw more valuable insights, qualitative studies from real-world organizational settings are vital.

REFERENCES

Billington, C., & Davidson, R. (2013). Leveraging open innovation using intermediary networks. Production and Operations Management, 22(6), 1464-1477.

Daniel, F., Kucherbaev, P., Cappiello, C., Benatallah, B., & Allahbakhsh, M. (2018). Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. ACM Computing Surveys (CSUR), 51(1), 1-40.

Donepudi, P. K. (2017). Machine Learning and Artificial Intelligence in Banking. *Engineering International*, *5*(2), 83-86. https://doi.org/10.18034/ei.v5i2.490

Foong, E., Gergle, D., & Gerber, E. M. (2017). Novice and Expert Sensemaking of Crowdsourced Design Feedback. Proceedings of the ACM on Human-Computer Interaction, 1(CSCW), 1-18.

Leicht, N., Blohm, I., & Leimeister, J. M. (2017). Leveraging the power of the crowd for software testing. IEEE Software, 34(2), 62-69.

Leicht, N., Knop, N., Blohm, I., Müller-Bloch, C., & Leimeister, J. M. (2016). When is crowdsourcing advantageous? The case of crowdsourced software testing.

Leicht, N., Rhyn, M., & Hansbauer, G. (2016). Can laymen outperform experts? The effects of user expertise and task design in crowdsourced Software Testing.

Liu, D., Bias, R. G., Lease, M., & Kuipers, R. (2012). Crowdsourcing for usability testing. Proceedings of the American Society for Information Science and Technology, 49(1), 1-10.

Muhdi L, Boutellier R (2011) the crowdsourcing process: an intermediary mediated idea generation approach in the early phase of innovation. Int J Entrepren Innov Manag 14(4):315–332. doi:10.1504/IJEIM.2011.043052

Stol, K. J., & Fitzgerald, B. (2014, May). Two's a company; three's a crowd: a case study of crowd sourcing software development, in Proceedings of the 36th International Conference on Software Engineering (pp. 187-198).

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.

--0--

How to Cite: