Pertanika J. Sci. & Technol. 25 (S): 27 - 38 (2017)

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Software Engineering in an Effective Collaborative Environment: An Evaluative Study on Crowdsourcing Platforms

Hani Al-Bloush[1]* and Badariah Solemon[2]

[1]*College of Graduate Studies, Universiti Tenaga Nasional, Selangor, Malaysia*
[2]*College of Computer Science and Information Technology, Universiti Tenaga Nasional, Selangor, Malaysia*

ABSTRACT

Crowdsourcing gathers the world's software engineering experts on a specific subject matter, and allows organisations and individuals to employ the combined effort of these 'experts' to accomplish the software task at hand. However, leveraging the knowledge of experts will not be achieved without online crowdsourcing platforms, which makes communication possible. This study intends to evaluate the performance of four Crowdsourced Software Engineering(CSE) platforms (TopCoder, InnoCentive, AMT and Upwork) based on the criteria of the Web of System Performance (WOSP) model. The WOSP criteria include functionality, usability, security, extendibility, reliability, flexibility, connectivity and privacy. Findings from the analyses showed that the four CSE platforms vary in all of their features, and at the same time, they all lack the requirements of flexibility. The results provide insight into the current status of CSE platforms and highlight the gaps inherent in these platforms while offering a more complete picture. This study contributes to work on enhancing the design of current and future platforms.

Keywords: Collaborative environment, crowdsourcing platform, crowdsourced software engineering (CSE), web of system performance (WOSP)

ARTICLE INFO

Article history:
Received: 09 March 2017
Accepted: 18 September 2017

E-mail addresses:
hanibarjas@yahoo.com (Hani Al-Bloush),
badariah@uniten.edu.my (Badariah Solemon)
*Corresponding Author

ISSN: 0128-7680 © 2017 Universiti Putra Malaysia Press.

INTRODUCTION

Crowdsourcing is now holding the world's attention, and has already revolutionised several aspects of human practice. The term 'crowdsourcing' was coined by Jeff Howe and Mark Robinson in Wired Magazine in June 2006 (Howe, 2006b). According to Howe (2006a) and Mao, Capra, Harman and Jia (2015), crowdsourcing is the act of organisations outsourcing their work to an undefined network of labourers, using an online open-call for participation. Currently, the online open-call communication process of crowdsourcing practitioners through platforms has caused great transformation in communication and networking, including TopCoder (Archak, 2010) and Amazon Mechanical Turk (AMT) (Begel, DeLine, & Zimmermann, 2010). These online platforms have made a quick move in various domains, among them involve the activities of software engineering. Software engineering is the application of engineering the development, design, implementation and maintenance of software (Laplante, 2007).

The last few years have seen great change in the manner by which software engineers develop their tasks at a high level of professionalism. This rise in professionalism lies behind the integrated series of software workers with assorted experience around the world, under the umbrella of crowdsourcing. In the world of software, crowdsourcing is growing quickly, and currently plays a critical role for most technology organisations (Dolstra, Vliegendhart, & Pouwelse, 2013). According to Mao et al. (2015), crowdsourcing has become a tool for development, testing and problem-solving in all software activities. This is referred to as "Crowdsourced Software Engineering (CSE)", and is a significantly effective tool. As stated in Stol and Fitzgerald (2014,pp.187-198), "Software engineering no longer takes place in small, isolated groups of developers, but increasingly takes place in organizations and communities involving many people."

Software organisations and individuals alike are using CSE platforms as an effective tool that has integrated software workers in order to meet software product requirements. Over the years, great contributions of online platforms have been made through many multinational organisations (e.g. IBM, L'Oracle, Dell) that offer crowdsourcing participation that focusses on several activities. This has prompted these companies to apply a crowdsourcing model to their global strategies (Li, 2016). The contribution of crowdsourcing to software engineering addresses several activities according to the software development life-cycle. Among these activities are: software requirements, design, coding, testing and verification and evolution and maintenance (Mao et al., 2015). The reasons behind the adoption of crowdsourcing by organisations and companies in their software activities are: easy access to a wide range of software workers, multiple solutions, low worker rates and reduced time-to-market (Kittur, Smus, Khamkar, & Kraut, 2011; Xiao & Paik, 2014).

This paper is an evaluative study on software crowdsourcing platforms that provide easy access to huge pools of software workers as an external solution provider. The study begins with a brief explanation of the selected CSE platforms, followed by a description of the Web of System Performance (WOSP) model and its criteria (Habbal, Chit, Ahmad, & Mahmod, 2015; Whitworth, Bañuls, Sylla, & Mahinda, 2008; Whitworth, Fjermestad, & Mahinda, 2006). Next, the details of the analysis of each criterion and a definitive review of the overall evaluation are presented. The conclusion of this study highlights the current case of online CSE platforms and proposes the required features to enhance current and future crowdsourcing platform designs.

Crowdsourced Software Engineering Platforms

Crowdsourcing platforms have received great attention through increasing demand for participation in the development of software engineering activities by users. Searches and practice show that crowdsourcing platforms are increasingly utilised by organisations for the development, designing, testing, evaluation and validation and maintenance of software applications, such as mobile applications, websites, accounting software, enterprise software and office suites (Zogaj, Bretschneider, & Leimeister, 2014).

Platforms vary from one to another in terms of task domains within software engineering context. Platforms such as TopCoder support multiple types of software engineering tasks such as coding, testing and designing (Mao, Yang, Li, & Harman, 2013). On the other hand, others are more particular such as uTest, which is designed for software testing (Gheorghe, 2015). Also, there are public platforms such as AMT and Inocentive, which are not designed for software engineering specifically, but can still be used to support multiple tasks for software engineering (Mao et al., 2015). As explained in LaToza and van der Hoek (2016), different platforms may use three different crowdsourcing models to process their individual tasks: Peer Production, Competition, and Microtasking. Peer Production contributes to software projects without managing and monetary rewards. This model has contributed to each of these projects: Wikipedia, Linux and Firefox (Wu, Tsai, & Li, 2013). As for the competition and microtasking models, both are similar in terms of monetary rewards, but differ in terms of task management. The competition model manages tasks through a series of competitions that require several days for completion. The other decomposes tasks into a set of self-contained microtasks that can be completed in a few minutes (Hoßfeld, Hirth, & Tran-Gia, 2011; LaToza, Towne, Adriano, & Van Der Hoek, 2014; Wu et al., 2013). It is clear from previous studies and the huge number of platform users that both competition and microtasking models have received broad adoption by commercial platforms. They have both helped achieve increasing speed, success and quality and have reduced the complexity of software engineering activities.

This paper evaluates four crowdsourcing platforms related to software engineering activities: TopCoder, InnoCentive, AMT and Upwork. These online crowdsourcing platforms were selected as they meet the following requirements:

- They offer support software engineering activities;
- They are a commercial platform; and
- They are accessible.

TopCoder

Topcoder is a company that manages competitions in software engineering activities. Topcoder conducts online competitions in programming known as Single Round Matches (SRMs) every two weeks. Also, it conducts weekly online competitions in graphic design and development. Design and development produce useful software that is licensed for profit by Topcoder.

Topcoder awards monetary prizes to winners and finalists of other weekly competitions or SRMs. It currently has one million registered users, who are both requesters and participants (Topcoder, 28 May, 2016).

InnoCentive. Innocentive is the global leader in open innovation, crowdsourcing and prize competitions. This company focusses on research and solving development problems in engineering, computer science, chemistry, maths, the life sciences, business and the physical sciences. Innocentive structures its activities as "challenge problems" for anyone willing to solve them. Cash rewards are offered for participants who meet the challenge criteria and offer the best solutions. The company recently announced significant traction for its Global Solver Community, with 300,000 individuals from more than 200 countries registered. More than 500,000 projects are carried out by the company (InnoCentive, 28 May 2016).

AMT. AMT is an online crowdsourcing platform for recruiting participants (called Providers in Mechanical Turk's Terms of Service or Turkers) to perform tasks known as Human Intelligence Tasks (HITs), posted by requesters. Participants can then navigate through the already existing tasks and complete them in exchange for a monetary reward set by the requester. The mechanism of paying participants for the tasks performed depends on the acceptance of the requesters. More than 500,000 participants from 190 countries are part of the company (Nichols, 21 Apr 2014).

Upwork. Upwork, formerly known as Elance-oDesk, is a global freelancing platform where software requesters can find software participants and collaborate remotely (Pofeldt, 5 May, 2015). In 2015, Elance-oDesk was rebranded as Upwork (Lunden, 5 May, 2015). Requesters of Upwork post a job and choose between hourly or fixed-price, then search for qualified participants and invite them to apply in order to complete their jobs. The mechanism of paying participants is done through hourly payment or fixed-price rates based on requesters' selections. Upwork has nine million registered freelancers and four million registered clients. Three million jobs are posted annually, equalling a total of $1 billion USD, making it a great online marketplace (Lunden, 5 May 2015).

The essential information provided by previous platforms is summarised in Table 1. A categorisation of essential information was based on Type, Industry, Foundation Year, Headquarters, Area Served, Key People and Website. The following information was derived from the company profile of each platform. Also included are some characteristics from early research on crowdsourcing platforms.

Table 1
Essential information – company profile

Essential Information	Crowdsourced Software Engineering Platforms			
	TopCoder	Upwork	AMT	InnoCentive
Type Industry	Subsidiary of Appirio Information technology staffing software outsourcing services	Private Freelance marketplace, Online outsourcing	Private Crowdsourcing Internet Marketplace	Private Crowdsourcing, Cloud labour, Open innovation, R&D, Innovation management, Product development
Founded	2001	2015	2005	2001
Headquarters	San Francisco, CA, USA	Mountain view, California	800 King Farm Boulevard, Rockville, Maryland	Waltham, Massachusetts, US
Area served	Worldwide	Worldwide	Worldwide	Worldwide
Key people	Jack Hughes	Sephane Kasriel and Thomas Layton	Mike Wiercinski	Craig Jones and Alpheus Bingham
Website	http://www.topcoder.com	http://www.upwork.com	http://www.mturk.com	http://www.innocentive.com

All the software crowdsourcing platforms described above were evaluated based on the WOSP criteria. The criteria were chosen due to their relevance to this research scope. The following section provides more details of the WOSP criteria.

Evaluation Criteria. The wosp model is an extension of the Technology Acceptance Model (TAM) (mahindra & whitworth, 2005; whitworth et al., 2008) approach by including measures based on the performance of the system, and is not based on user perception. WOSP is a theoretical framework utilised for balanced design and evaluation of advanced information systems. It analyses performance through four basic system elements: boundary, internal structure, effectors and receptors to define performance (whitworth et al., 2008). Each of these four fundamental elements can be designed to repel external threats, or to use external opportunities, contributing to eight performance goals. The eight performance goals could be categorised based on the four basic elements as effectors (functionality and usability), which have an effect on the environment and reduce action cost; boundary (security and extendibility), which is designed to prevent entry and outside objectives; structure (reliability and flexibility), which is designed for internal and external change; and receptors (connectivity and privacy), which enable and limit the ability to communicate. These criteria have been applied in the evaluation of malaysian crowdsourcing platforms conducted in 2015 (habbal et al., 2015). The wosp criteria and features used in this evaluation are listed in Table 2.

Evaluation Findings

The four platforms evaluated in this study can be classified according to their basic functionality and on whether they are dedicated platforms or public platforms. Dedicated platforms support multiple software development activities and tasks. As for public platforms are not designed for software engineering specifically, but can nevertheless be used to support various software development tasks. Each criterion was analysed based on the features related to the criterion.

Table 2
WOSP criteria (Habbal et al., 2015)

System Element	Evaluation Criteria	Description	Features	Testing
Effectors	Functionality	Ability to act effectively upon the environment	Basic functionality, Job matching (Job listings and seeking), Payment mechanism, Appraisal performance, Desktop application	Validity Testing: Input/Output Testbed
	Usability	Ability to operate efficiently or easily	Learnability, Efficiency (Task completion time), Navigation (broken links), Platform interface	Usability testing: Cognitive walkthrough, Protocol analysis, User heuristics.
	Security	A platform's ability to resist or avoid outside attack	Password, Random number, Captcha Image, Verification code	Penetration testing: Unauthorised entry probes, Attack tests
Boundary	Extendibility	Ability to use outside components or data	Openness (proprietary, tasks), Scalability, Interoperability, Compatibility (running on different browsers)	Compatibility testing: Plug-in testing, import/export testing
Internal and External Change	Reliability	Ability to avoid or recover from internal failure	Internal Failure (Errors), Availability.	Failure testing: Load tests, disaster recovery tests
	Flexibility	Ability to change to fit outer circumstances	Adaptability, Customisability, Modifiability	Situation testing: Languages, Users, Essential functions.
Receptors	Connectivity	Ability to communicate with other systems	Interactivity, Sociability, Communicativeness	Transmission testing: Test for sending/receiving information
	Privacy	Ability to control internal information	Confidentiality, Secrecy, Ownership rights (transfer and delegation of rights)	Social testing: Test whether the community accepts the system

All the four platforms provide basic functionality to support software engineering activities and job matching facilities for job listings and seeking. A payment mechanism is featured in all the platforms, but two of these platforms have pending obstacles (Upwork and AMT). An appraisal mechanism for workers and requesters is not featured in AMT. In terms of desktop applications, only Upwork provides this as an additional mechanism to remain connected with clients for communicating and tracking time.

In terms of usability evaluation, the majority of the platforms (three out of four platforms) are easy to learn and use by anyone who speaks English i.e. TopCoder, Upwork and InnoCentive. Tasks such as job seeking and posting can be accomplished easily, as they require only a few clicks. None of the platforms provide a multi-language facility, which allows users proficient in various tongues to use and participate on these platforms, according to *Upwork* users' discussions (Upwork, 4 April ,2016), as seen in the excerpt below:

> The client speaks my other native language so I have no problem communicating with them, but now they're asking me which buttons to press in order to make a job offer/pay me/etc. Is there any way for me as a freelancer to see what the client's page looks like, so I can look at it and explain to them in my language?

On all the platforms assessed, it was observed that users are required to register and use a password for logging in. However, for verification purposes, all the platforms send links via email in order to activate user accounts and to reduce the number of spam accounts, except for AMT. Only two out of four platforms (Upwork and InnoCentive) use a captcha image for greater security.

As for extendibility, all the platforms support openness and have a high degree of accessibility to view, utilise and contribute to modify the application. Only two platforms are scalable: TopCoder and AMT. TopCoder is enabled by appirio-tech and topcoder-UML-tool as modelling tools for use in design and development competitions. AMT is scalable by providing an on-demand scalable human workforce and gives workers a selection of thousands of tasks to complete whenever convenient (Mturk, n.d.). All the platforms are compatible with all main browsers and mobile devices except for *InnoCentive*, which encounters errors with Google Chrome and Microsoft Explorer. All the four platforms are interoperable by using a third-party web and application to support their services.

Reliability indicates the likelihood that the required function is performed without failure under the conditions laid down for a certain period of time. None of the evaluated platforms have internal failure, which can cause detrimental effects on system performance. In terms of availability, only InnoCentive performed well. TopCoder and Upwork could not be found all the time, with searches returning a 404 Error, while some pages of *AMT* contained broken links.

The flexibility of the platforms was assessed. None is adaptable, customisable and modifiable. They are not adaptable because they use only the English language, and do not allow foreigners a chance to use them even if adapted with multiple domains. They are not customisable because of the absence of a feedback mechanism. They are not modifiable because they do not allow the user to either change settings themselves or to request a change.

This study also evaluated the platforms' connectivity. Connectivity indicates their ability to interact with other users as well as to link to social media tools. All the platforms feature communicativeness. The study showed that all the platforms are linked to social media sites, except for AMT. Upwork is the only platform that is interactivity-enabled by desktop application to maintain data exchange.

Privacy ensures the transfer and delegation of submission rights and personal information that cannot be viewed by unauthorised parties. Confidentiality and secrecy are covered, and users are satisfied. However, transfer and delegation of submission rights are not addressed satisfactorily by the platforms, as shown by the evaluation and suggested by users' enquiries. Table 3 summarises the findings.

Table 3
Summary of analysis

	Commercial Platforms for Crowdsourced Software Engineering			
Criteria	Dedicated platforms		Public Platforms	
(√ denotes available)	TopCoder	Upwork	AMT	InnoCentive
Functionality				
Basic functionality	√	√	√	√
Job matching	√	√	√	√
Payment mechanism	√	Pending	Pending	√
Appraisal mechanism	√	√	None	√
Desktop application	None	None	None	√
Usability				
Learnability	√	√	None	√
Efficiency	√	√	None	None
Navigation	√	√	Broken links	√
Interface	English language only and not fit page edges	English language only	English language only, unsorted and complex	English language only and email text in invalid
Security				
Password	√	√	√	√
Random Number	None	None	None	None
Captcha Image	None	√	None	√
Verification Code	Verification email	Verification email	None	Verification email
Extendibility				
Openness	√	√	√	√
Scalability	√	None	√	None
Compatibility	√	√	√	√
Interoperability	√	√	√	√
Reliability				
Internal failure	√	√	√	√
Availability	√	√	None	None

Table 3 *(continue)*

Flexibility				
Adaptability	None	None	None	None
Customisability	None	None	None	None
Modifiability	None	None	None	None
Connectivity				
Interactivity	None	√	None	None
Communicativeness	√	√	√	√
Sociability	√	√	None	√
Privacy				
Confidentiality	√	√	√	√
Secrecy	√	√	√	√
Ownership rights	Worrying	Worrying	Worrying	Worrying

CONCLUSION AND FUTURE WORK

The findings of this study show that the four crowdsourcing platforms vary in all of their features, and at the same time, all of them lack flexibility. However, the application of the WOSP model in this study provided for more ideas on system performance in terms of reliability, flexibility and connectivity, among others. The findings revealed that all the platforms provide basic functionalities such as job posting and seeking and job matching. However, secured payment, appraisal mechanisms and keeping participants connected with clients for communicating and tracking time by using desktop applications are still lacking; these must be considered. Platforms should also feature strong confidentiality mechanisms to address user transfer and delegation rights. Additionally, flexibility is not given any emphasis. These platforms should also leverage from mobile and desktop applications to promote interaction in order to keep requesters and participants more connected. In order to guide future work in this area, we advocate for further research into understanding the roles played by the feedback mechanism and desktop application in allowing more engagement between software crowdsourcing practitioners and more customisation based on their requirements. Future work may also consider the concern of practitioners of intellectual property rights and the extent of their impact on participation in crowdsourced software engineering platforms. This would in turn allow for better crowdsourcing platform designs.

REFERENCES

Amazon Mechanical Turk. (n. d.). FAQs. Retrieved from https://www.mturk.com/mturk/help?helpPage=overview#what_is

Archak, N. (2010, April). Money, glory and cheap talk: analyzing strategic behavior of contestants in simultaneous crowdsourcing contests on TopCoder. com. In P*roceedings of the 19th international conference on World wide web* (pp. 21-30). ACM.

Begel, A., DeLine, R., & Zimmermann, T. (2010). Social media for software engineering. *Paper presented at the Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*.

Dolstra, E., Vliegendhart, R., & Pouwelse, J. (2013, March). Crowdsourcing gui tests. In *Software Testing, Verification and Validation (ICST), 2013 IEEE Sixth International Conference on* (pp. 332-341). IEEE.

Gheorghe, M. (2015). Software Quality Validation for Web Applications Developed Using Geographically Distributed Human Resources. *Informatica Economica, 19*(2), 101.

Habbal, A. M. M., Chit, S. C., Ahmad, R., & Mahmod, M. (2015). *Assessing Malaysian crowdsourcing platforms using web of system performance (WOSP) model*.

Hoßfeld, T., Hirth, M., & Tran-Gia, P. (2011, September). Modeling of crowdsourcing platforms and granularity of work organization in future internet. In *Proceedings of the 23rd International teletraffic congress* (pp. 142-149). International Teletraffic Congress.

Howe, J. (2006a). *Crowdsourcing: A definition.* Crowdsourcing: Tracking the rise of the amateur.

Howe, J. (2006b). The rise of crowdsourcing. *Wired Magazine, 14*(6),1-4.

InnoCentive. (2016, May 28). *InnoCentive Solver.* Retrieved from https://www.innocentive.com/innocentive-solver-network-passes-300000-registered-members-milestone/

Kittur, A., Smus, B., Khamkar, S., & Kraut, R. E. (2011, October). Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (pp. 43-52). ACM.

Laplante, P. A. (2007). *What every engineer should know about software engineering*. CRC Press.

LaToza, T. D., Towne, W. B., Adriano, C. M., & Van Der Hoek, A. (2014, October). Microtask programming: Building software with a crowd. In *Proceedings of the 27th annual ACM symposium on User interface software and technology* (pp. 43-54). ACM.

LaToza, T. D., & van der Hoek, A. (2016). Crowdsourcing in software engineering: Models, motivations, and challenges. *IEEE Software, 33*(1), 74-80.

Li, G. (2016). The Application and Innovation of Crowdsourcing in the Internet Age. *Open Journal of Social Sciences, 4*(03), 199.

Lunden, I. (2015, May 5). *Techcrunch.* Retrieved from http://techcrunch.com/

Mahindra, E., & Whitworth, B. (2005). *The web of system performance: Extending the TAM model.* AMCIS 2005 Proceedings, 173.

Mao, K., Capra, L., Harman, M., & Jia, Y. (2015). A survey of the use of crowdsourcing in software engineering. *RN, 15*(01).

Mao, K., Yang, Y., Li, M., & Harman, M. (2013, May). Pricing crowdsourcing-based software development tasks. In *Proceedings of the 2013 international conference on Software engineering* (pp. 1205-1208). IEEE Press.

Nichols, I. (2014, April 21). *Amazon web services*. Retrieved from https://forums.aws.amazon.com/thread.jspa?threadID=150722

Pofeldt, E. (2015, May 5). *Elance-oDesk becomes 'upwork' in push to build $10B in freelancer revenues*. Retrieved from http://www.forbes.com/sites/elainepofeldt/2015/05/05/elance-odesk-becomes-upwork-today-odesk-brand-gets-phased-out/#6ccd49672bed

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

Topcoder. (2016, May 28). *TopCoder.* Retrieved from https://www.topcoder.com/

Upwork. (2016, April 4). *Community discussions.* Retrieved from https://community.upwork.com/t5/Freelancers/How-can-I-help-a-client-who-barely-understands-English-to/td-p/175911/

Whitworth, B., Bañuls, V., Sylla, C., & Mahinda, E. (2008). Expanding the criteria for evaluating socio-technical software. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 38*(4), 777-790.

Whitworth, B., Fjermestad, J., & Mahinda, E. (2006). The web of system performance. *Communications of the ACM, 49*(5), 92-99.

Wu, W., Tsai, W. T., & Li, W. (2013). Creative software crowdsourcing: from components and algorithm development to project concept formations. *International Journal of Creative Computing, 1*(1), 57-91.

Xiao, L., & Paik, H. Y. (2014, April). Supporting complex work in crowdsourcing platforms: A view from service-oriented computing. In *Software Engineering Conference (ASWEC), 2014 23rd Australian* (pp. 11-14). IEEE.

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.