Process Improvements for Crowdsourced Software Testing

Sultan Alyahya, Dalal Alrugebh
Information Systems Department
King Saud University, Riyadh, Saudi Arabia

***Abstract*—Crowdsourced software testing has been a common practice lately. It refers to the use of crowdsourcing in software testing activities. Although the crowd testing is a collaborative process by nature, there is no available research that provides a critical assessment of the key collaboration activities offered by the current crowdsourced testing platforms. In this paper, we review the process used in the crowd testing platforms including identifying the workflow used in managing the crowd testing process starting from submitting testing requirements and ending with reviewing testing report. Understanding the current process is then utilized to identify a set of limitations in the current process and has led to propose three process improvements (improving assigning crowd manager, improving building test team, monitoring testing progress). We have designed and implemented these process improvements and then evaluated them using two techniques: 1) questionnaire and 2) workshop. The questionnaire shows that the process improvements are significantly sound and strong enough to be added to crowd testing platforms. In addition, the evaluation through conducting a workshop was useful to assess the design and implementation of the process improvements. The participants were satisfied with them but asked for further modifications. Nevertheless, because crowd testing requires participation from a large number of people, the automation suggested improving managing the current process which was highly appreciated.**

***Keywords*—Software testing; crowdsourcing, crowd testing; process improvement; tool**

I. Introduction

In the last few years, a recent software testing strategy called crowdsourced testing has become a common practice and has been increasingly introduced into software development organizations. Crowdsourced testing refers to the use of crowdsourcing in software testing activities. It differs from the traditional approach; in this, testing is carried out by a larger number of testers from different places as crowd test team which can be consisted of professional testers, novice testers, real application users and subject matter experts, rather than by a limited number of in-house testing professionals [1].

According to Bruce Sterlings [2]: "*The best software is that which has been tested by thousands of users under thousands of different conditions, over the years*". This is almost impossible with the classical testing approach.

Crowdsourced testing has three main components [3]:

1) *The crowd testers*: the individuals who carry out the testing.

2) *The crowd seekers*: project owners who submit the projects for testing.

3) *An intermediation platform*: building a link between crowd testers and crowd seekers. This serves as a crowdsourcing enabler that allows customers to express their needs and individuals making up the crowd to respond to these needs.

There are many research papers that explore the use of crowdsourced testing platforms in the various types of testing such as usability testing [4], performance testing [5], GUI testing [6], etc.

However, although the crowd testing is a collaborative process by nature, there is no available research that provides a critical assessment of the key collaboration activities offered by the current crowdsourced testing platforms. Mao *et al*. [7] mention that the coordination and communication issue in the current platforms has not been explored yet. They state:

> *"both the resources and development process need to be coordinated. For example, geographically distributed and transient crowd workers need to reach a consistent understanding of the tasks required of them. Without coordination, it may be quite problematic".*

In this paper, we will review the process used in the crowdsourced testing platforms including identifying the workflow used in managing the crowd testing process starting from submitting testing requirements and ending with reviewing testing report. Understanding the process used is the first step towards assessing the current platforms and then providing process improvements for them.

The remainder of this paper is organized as: Section II provides some background information about the crowdsourced testing and discusses the related work. Section III describes the current crowdsourced testing process. Observations on the current process are discussed in Section IV while process improvements are proposed in Section V and the evaluation is presented in Section VI. Their implementation is shown in Section VII and finally, the paper concludes in Section VIII.

II. Background and Related Work

A. Crowdsourced Software Testing

Crowdsourced software testing (also known as 'crowdsourced testing' and 'crowd testing') is to outsource testing activities to testers recruited from a large pool of individuals. It overcomes the limitations of the conventional in-house testing that is restricted to the knowledge of a small set of solvers and thus is limited in terms of quality and efficiency [8].

Crowd testing is not a replacement for conventional testing, but a supplement solution done by people who are not directly involved in the project. They can be from different geographical and cultural backgrounds performing exploratory testing, identify defects, and provide user experience feedback.

Crowd testing is successfully implemented by using crowdsourcing platforms. These platforms play a key role in managing the process of crowd testing as will be clarified in Section III.

TABLE. I. CROWD SOURCED TESTING PLATFORMS

Application Name	**Types of Tests Offered**
Utest	• Functional • Security • Usability • Load • Localization
Crowdsourced Testing	• Functional • Usability • Localization
Pay4Bugs	• Functional • Localization
Mob4Hire	• Usability • Functional
99tests	• Functional • Security • Load • Automation
Passbrains	• Functional • Compatibility • Security • Localization • Usability
TestCloud	• Functional
Feedback Army	• Usability

uTest [9] claims to be the largest community for software testing. In addition to this one, the most discussed crowd testing platforms in the literature includes crowdsourced testing [10], Pay4bugs [11], Mobile4Hire [12], 99tests [13], Passbrains [14], TestCloud [15] and Feedback Army [16] (shutdown October 2016). Table 1 lists the types of testing offered by these platforms.

The role of each member in crowd testing platforms is described below:

1) *Crowd Testing Manager*: Crowd testing manager is a person who manages the community of the crowdsourced software testing, including building test teams and providing learning and engagement experiences for crowd testers.

2) *Test Team Leader*: Test team leader is responsible for the performance and management of a group of crowd testers in a given project and ensuring that the needs of the crowd seekers are met. The test team leader is also the escalation point for all bugs and technical fault.

3) *Crowd Testers*: Crowd testers are the heart line of crowdsourced software testing to execute testing activities. Crowd testers can be from different levels as well as from domain experts or potential users.

In current crowd testing platforms, each crowd tester has a profile with basic information and project preferences. The basic information is about crowd testers' demographic information and their experience with several testing types (functional, load, etc.), operating system, test automated tools and hardware [3].

4) *Crowd Seekers (Customer)*: Crowd seeker is the project owner who crowdsources the testing activities to the crowd.

B. Related Work

The use of crowdsourced testing has been studied with several types of testing activities, such as usability testing, performance testing and GUI testing.

With usability testing, the existing work attempts to evaluate the crowdsourced usability testing against the traditional approach in terms of its capability for detecting usability problems [4].

Musson *et al.* [5] demonstrate the value of using crowd testing to measure real-world performance and how it can mitigate the problem of finding various user behaviors and execution environments for testing software in various regions in the world.

Dlstra *et al.* [6] consider using crowd testing to make continuous GUI testing. They have described a method for crowdsourcing of GUI tests based on instantiating the system under test in virtual machines that are served by a geographically dispersed pool of workers.

The literature discussed above aim to evaluate the suitability of using crowdsource concept in testing activities. However, they do not consider making an assessment to the current crowdsourced testing process.

Furthermore, many research papers have focused on developing a theoretical basis for the crowdsourced software engineering in general. The authors in [17] propose Metropolis Model. They argue that the classical software development models (e.g. waterfall, agile models) are not suitable for Crowdsourced Software Engineering. Tsai *et al.* [18] summarized the commonalities in different Crowdsourced Software Engineering processes and proposed an architecture for cloud-based software crowdsourcing.

In addition to the fact that these papers do not concentrate on crowd testing in particular, they merely provide generic-level frameworks to use crowdsourcing in software engineering without any assessment of the process used in the current crowdsourcing platforms.

III. Current Crowd Sourced Testing Process

The platforms listed in Table 1 have been reviewed to understand the current process of crowd testing. The current crowd testing platforms share the same common workflow shown in Fig. 1. We considered making high granularity of analysis in order to provide a unified workflow that all the crowd testing platforms meet.

In order to provide a comprehensive review of the current platforms, we have used the following methods:

1) Registering and participating in the surveyed platforms (i.e. mainly as crowd testers).

2) Watching demos explaining the platforms' functionalities.

3) Reading the formal descriptions of the platforms.

4) Asking direct questions through relevant community boards.

The activities used in these platforms are described below:

a) *Submit the project*: The crowd seekers submit software project for testing and specify the business needs and goals clearly. At this stage, the crowd seeker needs to submit: the targeted software, required testing types, required operating system platforms and required testing automation tools.

b) *Review the submitted project by crowd manager*: The crowd manager reviews the project based on the requirements of the crowd seeker, crowd manager designs test plan that includes estimation of testing size, testing effort and testing cost.

c) *Announce the project*: After the crowd manager reviews the submitted project, an announcement will be sent to the crowd testers based on: their performance (i.e., quantity and quality of bugs they submit) and the frequency of participation in test cycles. In addition, there are many factors go into test cycle invitations such as the crowd's available hardware, software and geography location.

d) *Review project specification by crowd testers*: When crowd testers receive a test cycle invitation, they will have the option to either accept or decline the project opportunity. They must accept a test cycle to be able to report bugs and test cases. On the other hand, if they are not available for testing or perhaps they are not interested in a particular test cycle, they have the option to decline the invitation.

If the invitation is ignored, this will impact negatively on the crowd testers' rating. It is also important that crowd testers review all instructions and documentation before they start testing to avoid "out of scope" bugs, which can lower crowd testers' rating too.

e) *Build test team*: The crowd manger reviews the list of crowd testers interested to work in the test cycle. He verifies their information and makes sure they satisfy the required qualifications to participate.

The crowd manger recruits crowd testers based on matching customer requirements of hardware and software with crowdsourced testers' evaluation, available resources and demographic information.

After crowd testers are selected based on the review, they are then presented with the test cases associated with a particular test cycle. Under "Available Test Cases", crowd tester can view which test cases are available to claim based on testing environment and pre-determined rules set by the crowd seeker (e.g., specific number of testers per environment, specific number of pending test cases per tester).

f) *Assign crowd test team leader*: Test team leader is selected by crowd manager to work under the guidance of a crowd manager. For each test cycle, the leader recommends approval/rejection of incidents to increase the value of the final report (i.e., by improving the output and minimizing the noise level).

g) *Test the project*: In this activity, crowd testers start testing using the required types of devices, operating systems and software.

h) *Submit incident testing report*: It is the responsibility of the crowd tester to create an incident, link it to a corresponding script, and assign an initial severity and status.

The incidents are composed of the following:

i) Incident title and description of the issue.

ii) Actions and steps performed, numbered and clearly written, showing how the user arrived at the bug location.

iii) Environment information containing information, such as the operating system used browser and URL from where the bug is found.

iv) Attachments showing the issue either in picture or in video.

i) *Evaluate the incidents by crowd leader*: It is the responsibility of the crowd leader to review the severity of the incidents and modify status as incidents progress through the test cycle. The incidents may be pushed to one of two other states called "Pending Approval" or "Pending Rejection".

j) *Update crowd testers rating by crowd manager*: Based on incident reports evaluation, these reports may have a positive impact on crowd testers rating if the identified bugs are classified as "exceptionally valuable", "very valuable" or "somewhat valuable". On the other side, "rejected bugs" have a standard negative impact on crowd testers' rating.

k) *Prepare testing reports*: At this step, depending on the crowd seeker's feedback, the incidents may be moved into "Approved" or "Rejected". In addition, experience and lesson learned are documented and notifications are sent to close out any activities related to the project.

l) *Review final report*: Incidents exported, tester feedbacks, test summary and recommendations are handed to customer for final review.

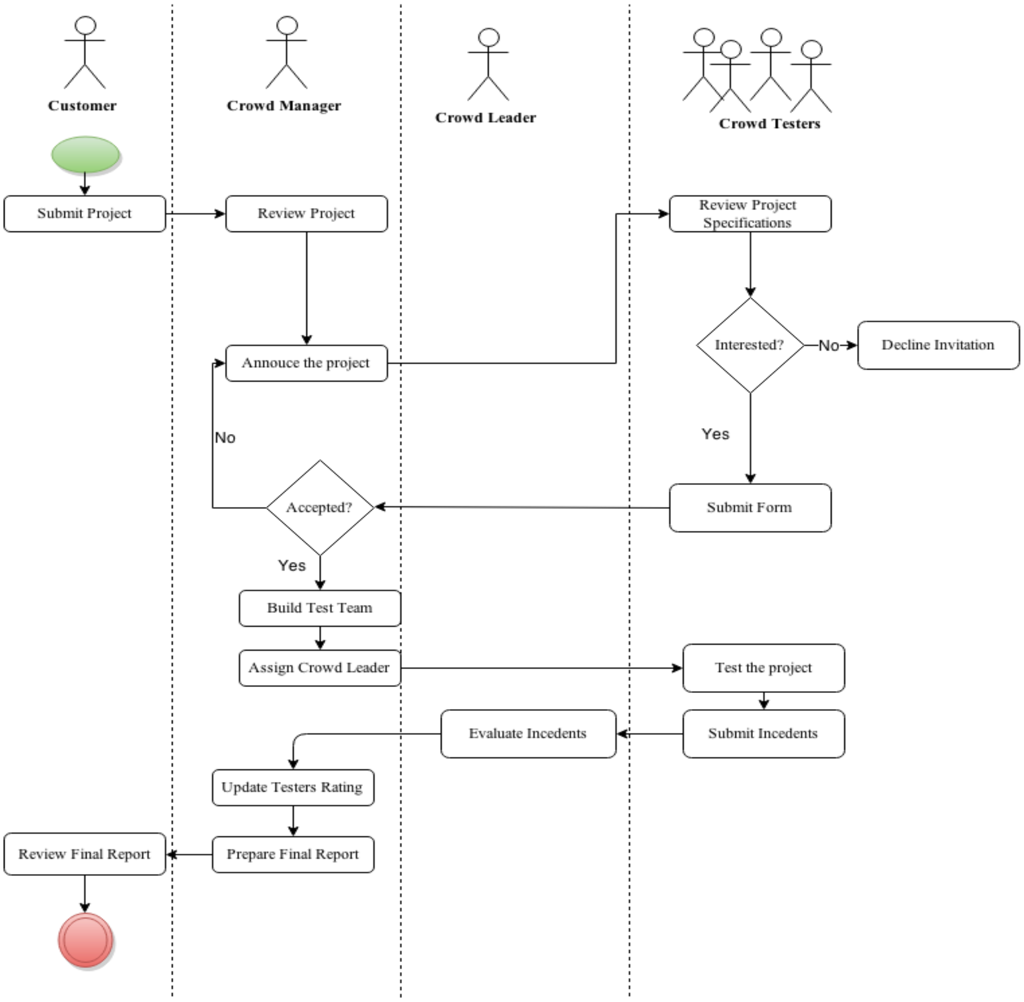


Fig. 1. Crowdsourced software testing workflow.

IV. Observations on the Current Process

The study of the current process used in crowdsourced testing reveals several observations that we believe can be source of weaknesses. These observations are:

1) Crowd mangers are assigned manually. In crowdsourced testing, it is common that multiple projects share the same crowd managers; hence, with the manual process used to assign crowd managers, the allocation can be inefficient (i.e., some crowd managers can overwork while others wait for new assignments).

2) Building the testing team is carried out manually. Reviewing the list of interested crowd testers and matching their capabilities with the test requirements are time consuming specially for large projects (i.e., some projects can receive hundreds of candidate participants).

3) There is a lack of mechanisms for monitoring the progress of crowd testers. They may accept to participate but then disappear or do not submit any test report.

V. Proposed Process Improvements

Based on the process observations listed in Section IV, we propose in this section a set of process improvements to be implemented in a crowdsourced testing computer application (CSTCA). These are listed below:

Process Improvement 1: Improving Assigning Crowd Manager

A resource allocation process is needed to allocate projects to the free crowd managers (or the least loaded). In our proposed process improvement, an automatic assignment of project crowd is offered based on their availability and preferences in testing types or testing environments. The most available one will be the primary crowd manager of a new project.

Fig. 2 presents the process of assigning crowd manager. After crowd seeker submits a software project with an estimate of the expected date of delivery, the system will send automatic notification to crowd managers based on testing requirements (e.g., testing type/environment). If a crowd manager accepts the project, they should provide an estimate about when they will be available to start testing the project. The system then assigns a crowd manager based on the best availability.

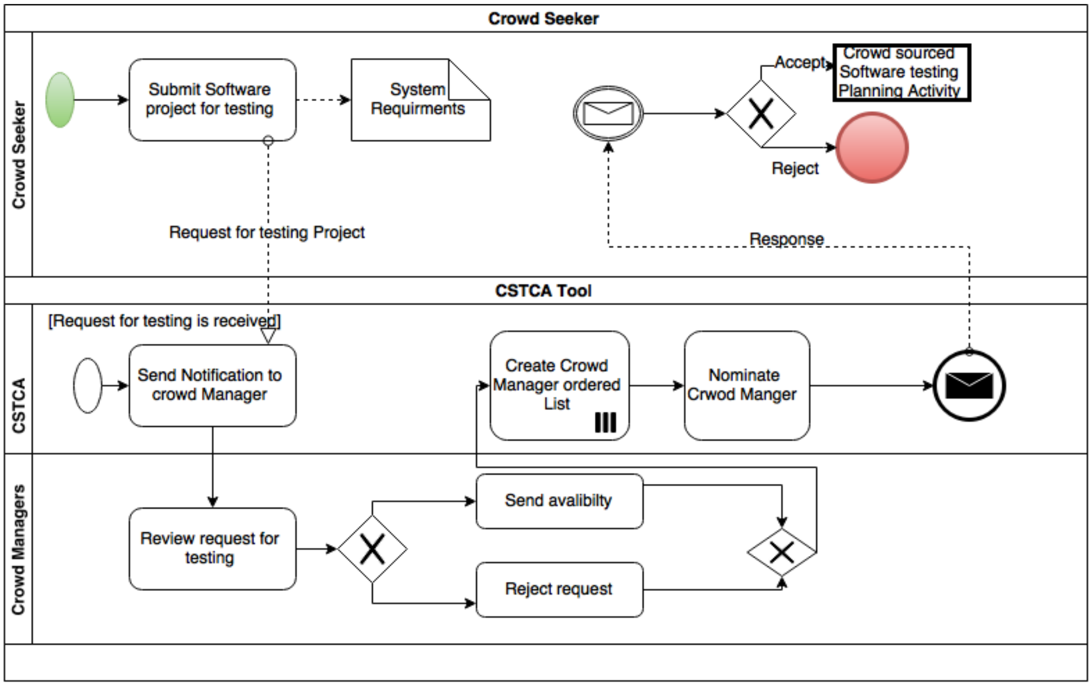

Crowd Seeker
Crowd Seeker
Submit Software project for testing
System Requirments
Accept
Crowd sourced Software testing Planning Activity
Reject
Request for testing Project
Response
CSTCA Tool
CSTCA
[Request for testing is received]
Send Notification to crowd Manager
Create Crowd Manager ordered List
Nominate Crwod Manger
Crowd Managers
Review request for testing
Send avalibilty
Reject request

Fig. 2. The workflow of the proposed Process Improvement 1.

Process Improvement 2: Improving Building Test Team

To avoid the limitations of the manual building of a test team, the system will pick up the most relevant crowd testers automatically.

Fig. 3 presents this process improvement. The system announces the project for those who meet the project requirements. After crowd testers register for a project, the system will accept the maximum allowed number of crowd testers based on crowd testers' availability and rate.

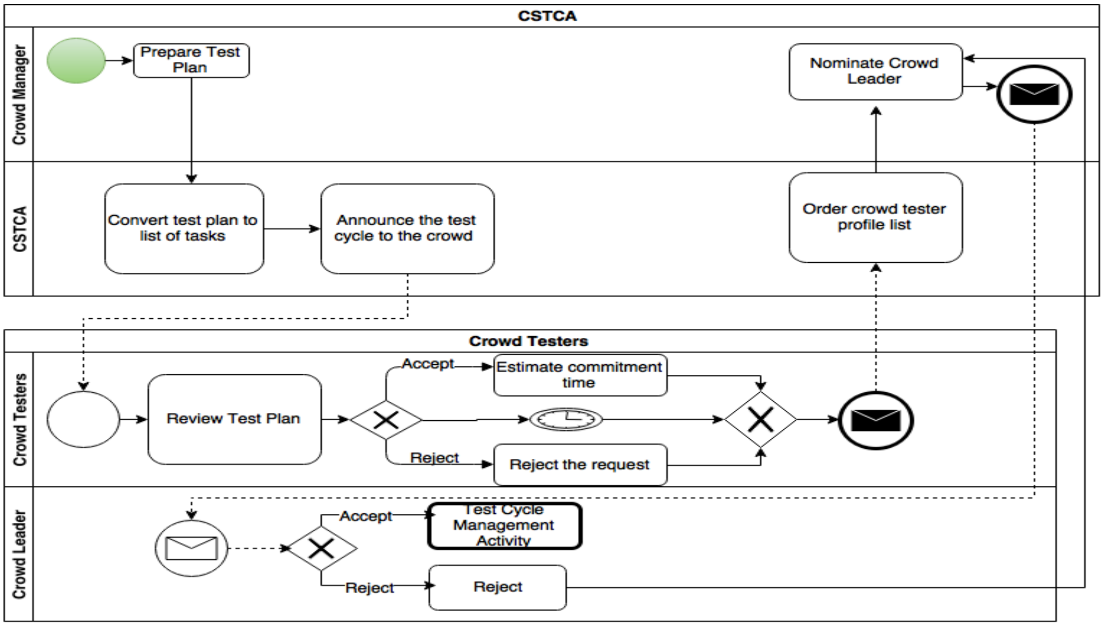


Fig. 3. The workflow of the proposed Process Improvement 2.

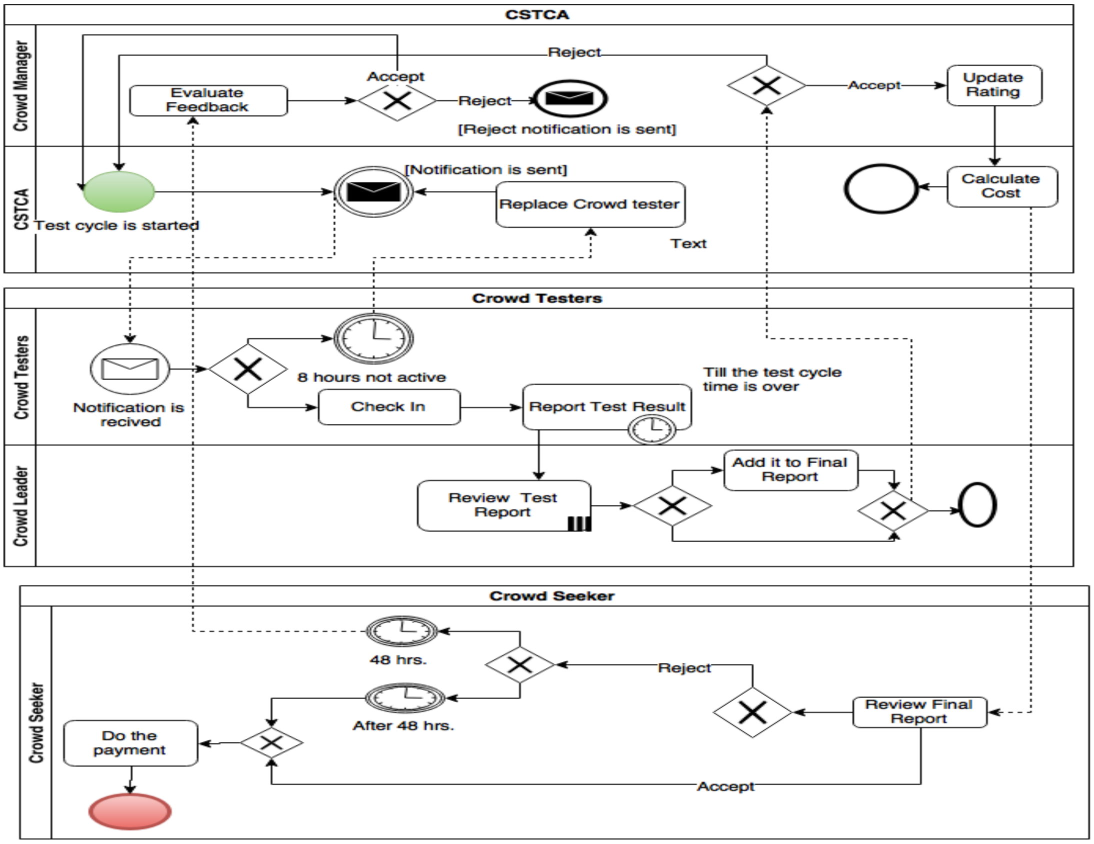


Fig. 4. The workflow of the proposed Process Improvement 3.

Process Improvement 3: Monitoring Testing Progress

To minimize the problem of having non-productive crowd testers, a process improvement is proposed to monitor the work of crowd testers and flag those who are not active.

Fig. 4 shows that if the crowd tester does not login or be active for a specific period of a work day (e.g., 8 hours), the system will exclude him and send invitation to other crowd testers.

The process can encourage crowd testers to accept a test cycle invitation only if they are willing to spend appropriate amount of time to accomplish the assigned test cases.

VI. Implementation

As a proof of concept, we have developed a research-based crowdsourced testing computer application (CSTCA).

CSTCA is designed as multi-layer architecture as shown in Fig. 5. The layers of the architecture are the Presentation Tier (i.e., User Interfaces), Domain Tier and Data Tier (i.e., Database).

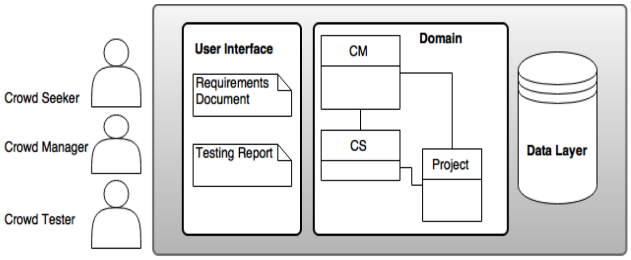
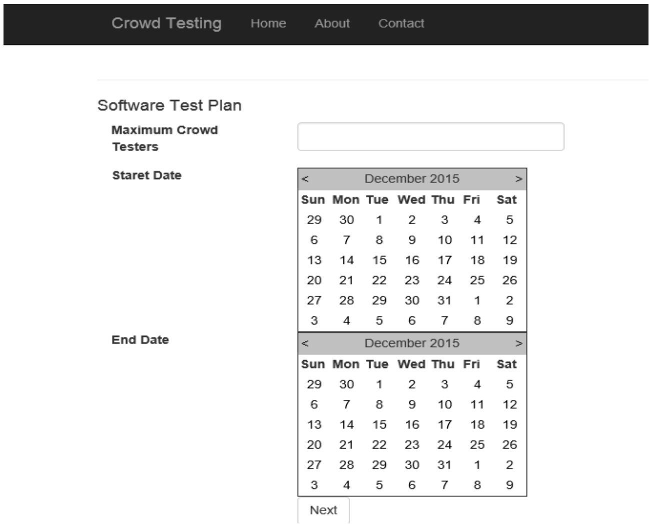


Fig. 5. The architecture of CSTCA.

For the sake of brevity, the implementation of the second process improvement (Improving Building Test Team) is only shown here. This process starts by preparing a test plan by a project plan. Fig. 6 shows the screen of preparing test plan. The test plan will include the estimation of minimum number of crowd testers should be participated in a test cycle with the start and end dates of the test cycle.



Software Test Plan

Maximum Crowd Testers

Staret Date

< December 2015 >

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

End Date

< December 2015 >

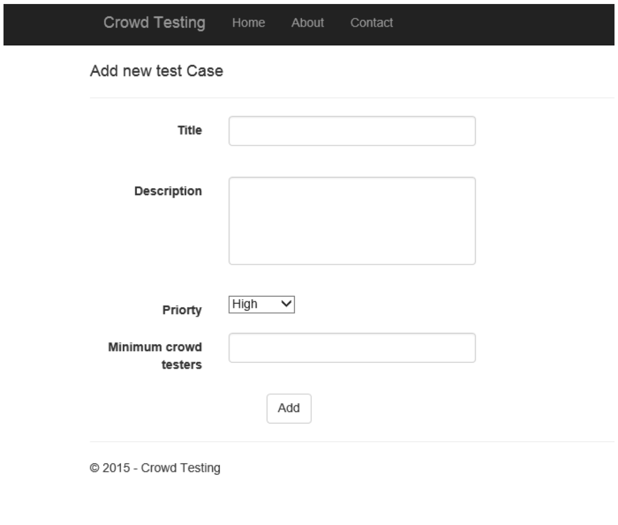
Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Fig. 6. Preparing the test plan.

The test plan includes a list of test cases to be performed by crowd testers. Fig. 7 shows the screen of adding test cases. The test case should include test case title, description, priority and number of minimum crowd testers needed.

Once the test cycle is created by crowd manger, notifications will be sent automatically to crowd testers based on their availability, rate and their preferences in testing types or testing environments.

The crowd testers have the option to accept or reject an invitation. If the invitation is accepted by a crowd tester, he will be asked to estimate the total number of hours he can spend in the test cycle as shown in Fig. 8.



Add new test Case

Title

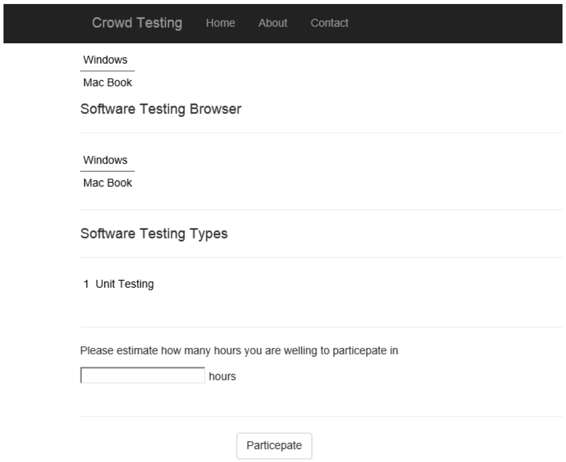
Description

Priorty High

Minimum crowd testers

© 2015 - Crowd Testing

Fig. 7. Test case form.



Windows
Mac Book

Software Testing Browser

Windows
Mac Book

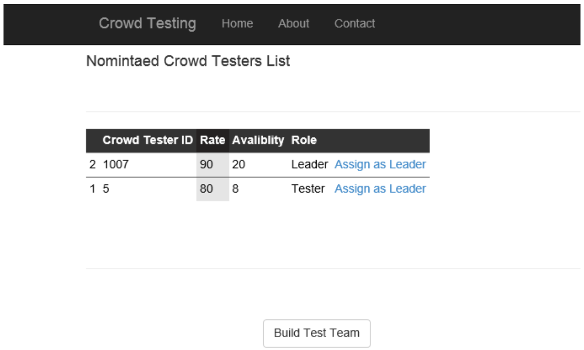
Software Testing Types

1 Unit Testing

Please estimate how many hours you are welling to particepate in

hours

Fig. 8. Reviewing project details by crowd tester.



Nomintaed Crowd Testers List

	Crowd Tester ID	Rate	Avaliblity	Role	
2	1007	90	20	Leader	Assign as Leader
1	5	80	8	Tester	Assign as Leader

Fig. 9. Testing team is built.

The crowd manger will have the option to build test team by one click. The system will select the crowd testers based on their availability, rate and testing type preferences. The crowd leader will be assigned automatically based on his rate. To provide more flexibility, the crowd manger will have the option to change the crowd leader if needed as shown in Fig. 9.

VII. EVALUATION

This section details the evaluation of the proposed process improvements. A combination of questionnaire and workshop methods is used for the evaluation.

The main purpose of the questionnaire is to make sure that the identified process improvements are sound and strong enough to be added as part of the crowdsourced software testing process. Furthermore, the main purpose of the workshop is to go deeply inside the new processes and identify any limitations which can affect the applicability of them.

A. Questionnaire

Although it seems to us that there is some value in introducing the three suggested process improvements, we believe it is necessary to ask people who actually work in the problem domain about their opinions regarding the need for these process improvements.

We have selected twelve domain experts to answer a short questionnaire. It is taken into account that the selected people must have strong background about the domain of crowd testing. It also considered that they have diverse roles as shown in Fig. 10 (i.e., includes people with the roles: crowd seeker, crowd leader, crowd manager and crowd testers).

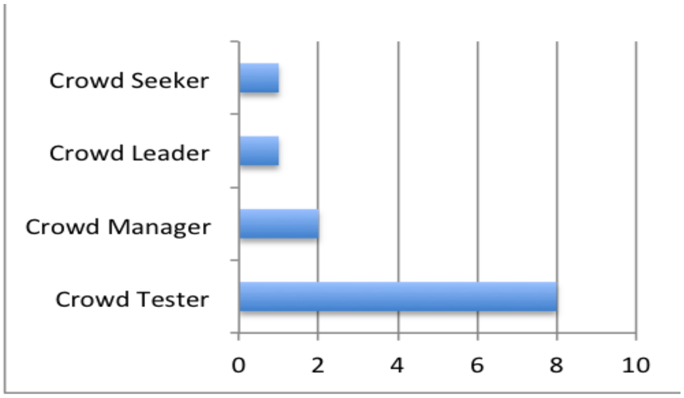


Fig. 10. Roles of the domain experts.

Feedback about Process Improvement 1:

We asked the domain experts to evaluate the need for improving the process of identifying the right crowd manager from their experience. The feedback is shown in Fig. 11. Two third of them agree that this improvement will be useful.

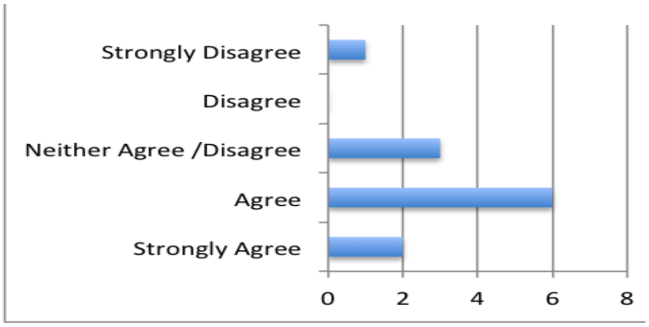


Fig. 11. Results of evaluating Process Improvement 1.

Feedback about Process Improvement 2:

The domain experts have evaluated the process of automating building test teams (Fig. 12). Ten out of twelve agree that this improvement will be useful.

Feedback about Process Improvement 3:

The idea of monitoring the testing process has also been evaluated by the domain experts as shown in Fig. 13. Again here, two third of them agree that this improvement will be useful.

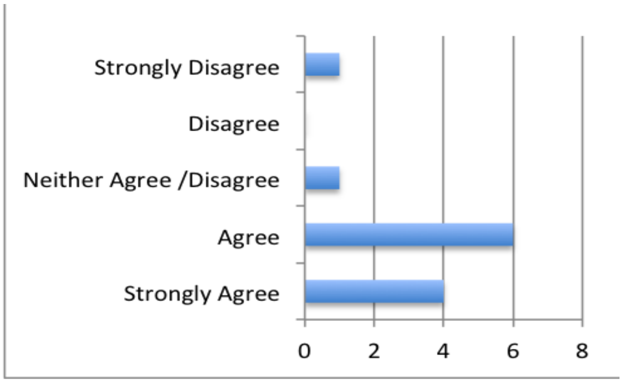


Fig. 12. Results of evaluating Process Improvement 2.

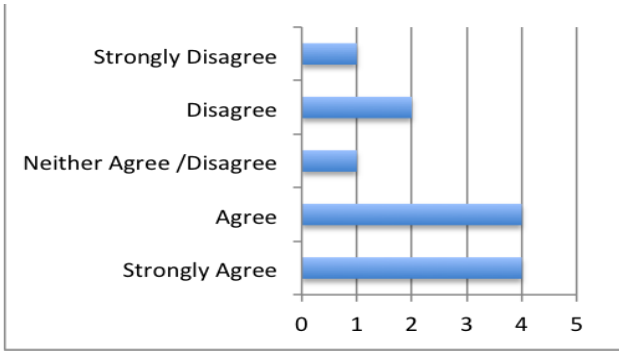


Fig. 13. Results of evaluating Process Improvement 3.

As we mentioned earlier, the purpose of the questionnaire is to have general view about the validity of the process improvements we suggested. It is clear from the results that the majority of the domain experts believe that the process improvements are useful. The results were encouraging for us to move to the stage of designing and implementing the proposed process improvements.

B. Workshop

Workshop Setup

A technical workshop is carried out to evaluate the design and implementation of the new process improvements. Two software process engineers and four domain experts have been invited for the workshop. They have been handed a handbook describing the process design and implementation in advance. The workshop is set to complete in one hour and a half so that each process improvement takes about half an hour. The room is equipped with a data show to provide live demo to the attendees.

Workshop Session

With each process improvement, the moderator has explained in details the proposed process workflow and displayed a demo about the corresponding feature. Afterwards, questions were received to make sure that the participants eliminate any ambiguity about the proposed process improvements. Then, the workshop discussion started. The theme of the discussion focus on three main questions directed to the participants:

1) What is your opinion about the design and

implementation provided to improve the process of assigning crowd manager?

2) What is your opinion about the design and implementation provided to improve the process of building test team?

3) What is your opinion about the design and implementation provided to improve the process of monitoring testing progress?

Workshop Findings

The workshop session was useful to assess the proposed process improvements. The participants believe that the modifications carried out will improve the current process of crowd testing. All the participants appreciate the automation part of the process improvement and they believe that it will add value to the crowd testing platforms. Examples of their words are:

"Rather than waiting for project managers to decide on which one should take the responsibility of a new project, why do not we set criteria for the selection and make that automatic … it speeds up the process."

"We send invitation to all testers, it takes time to filter the responses and decide who should be involved … automatic creation of test team can be great feature".

Furthermore, for the workflow of Process Improvement 1 (Fig. 2), the participants believe that, instead of asking project managers about their availability in the mid of the assignment process, it would be better that they register their availability in advance. This can shorten the assignment process.

In addition, with regard to the workflow of Process Improvement 2 (Fig. 3), the participants mentioned that it is necessary to avoid the full automation of creating a test team. They suggested that the project manager should review the names of the nominated crowd testers after it is automatically identified by the system. They believe that many human-based factors can determine the relationship between crowd managers and crowd testers especially if they have experience of working together in old projects (e.g., socio-cultural factors, trust, etc.) which can affect the success of the test project.

Finally, the participants stated that the third process improvement (Fig. 4) will solve the problem of having many testers accept test invitation but then disappear. However, they suggested relaxing the condition that judges whether a crowd tester is productive in a project or not. They argue that it is the crowd manager who should determine first the acceptable amount of time allowed before excluding a crowd tester; then the system can automate the exclusion task. They consider the length of project and the planned completion date of project as important factors in this matter.

VIII. Conclusions

This paper narrows the research gap regarding studying crowd testing process. It reviewed the current workflow used in crowd testing platforms and proposed three process improvements.

The questionnaire shows that the process improvements are significantly sound and strong enough to be added to crowd testing platforms. In addition, the evaluation through conducting a workshop was useful to assess the design and implementation of the process improvements. The participants were satisfied with them but asked for further modifications. Nevertheless, because crowd testing requires participation from large number of people, the automation suggested to improve managing the current process was highly appreciated.

This research focuses only on crowd testing. Future research will investigate examining the process of using crowdsourcing concept in other software engineering phases, such as requirements engineering.

References

[1] D. Speidel and M. Sridharan, "A framework and research agenda for crowdsourced testing," 2013.

[2] B. Zajac, "The Hacker Crackdown," *Comput. Fraud* , vol. 1994, pp. 18–19, 1994.

[3] M. Sharma and R. Padmanaban, *Leveraging the Wisdom of the Crowd in Software Testing*. CRC Press, 2014.

[4] C. Schneider and T. Cheung, "The power of the crowd: Performing usability testing using an on-demand workforce," in *Information Systems Development*, Springer, 2013, pp. 551–560.

[5] R. Musson, J. Richards, D. Fisher, C. Bird, B. Bussone, and S. Ganguly, "Leveraging the crowd: How 48,000 users helped improve Lync performance," *IEEE Softw.*, vol. 30, no. 4, pp. 38–45, 2013.

[6] E. Dolstra, R. Vliegendhart, and J. Pouwelse, "Crowdsourcing GUI tests," in *Proceedings - IEEE 6th International Conference on Software Testing, Verification and Validation, ICST 2013*, 2013, pp. 332–341.

[7] K. Mao, L. Capra, M. Harman, and Y. Jia, "A Survey of the Use of Crowdsourcing in Software Engineering," *RN*, vol. 15, p. 1, 2015.

[8] S. Zogaj, U. Bretschneider, and J. M. Leimeister, "Managing crowdsourced software testing: a case study based insight on the challenges of a crowdsourcing intermediary," *J. Bus. Econ.*, vol. 84, no. 3, pp. 375–405, 2014.

[9] "uTest." [Online]. Available: https://www.utest.com/. [Accessed: 01-Dec-2016].

[10] "Crowdsourced Testing." [Online]. Available: https://crowdsourcedtesting.com. [Accessed: 01-Dec-2016].

[11] "Pay4bugs." [Online]. Available: https://www.pay4bugs.com/. [Accessed: 01-Dec-2016].

[12] "Mobile4Hire." [Online]. Available: http://www.mob4hire.com/. [Accessed: 01-Dec-2016].

[13] "99tests." [Online]. Available: https://99tests.com/. [Accessed: 01-Dec-2016].

[14] "Passbrains." [Online]. Available: https://www.passbrains.com/. [Accessed: 01-Dec-2016].

[15] "TestCloud." [Online]. Available: https://www.xamarin.com/test-cloud.

[16] "Feedback Army." [Online]. Available: http://feedbackarmy.com/. [Accessed: 01-Dec-2016].

[17] R. Kazman and H.-M. Chen, "The metropolis model a new logic for development of crowdsourced systems," *Commun. ACM*, vol. 52, no. 7, p. 76, 2009.

[18] W. T. Tsai, W. Wu, and M. N. Huhns, "Cloud-based software crowdsourcing," *IEEE Internet Computing*, vol. 18, no. 3. Institute of Electrical and Electronics Engineers Inc., pp. 78–83, 2014.