JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 38, 715-733 (2022)
DOI: 10.6688/JISE.202207_38(4).0001

Trust Issues in Crowdsourced Software Engineering: An Empirical Study

HUMA HAYAT KHAN[1], MUHAMMAD NOMAN MALIK[2,+]
AND YOUSEEF ALOTAIBI[3]
[1]*Department of Software Engineering*
[2]*Department of Computer Science*
[1,2]*Faculty of Engineering and Computer Sciences*
National University of Modern Languages
Islamabad, 44000 Pakistan
[3]*Department of Computer Science*
College of Computer and Information Systems
Umm Al-Qura University
Makkah, 21955 Saudi Arabia
E-mail: {hanuman; mnauman}@numl.edu.pk; yaotaibi@uqu.edu.sa

Software crowdsourced has emerged in recent years, and is an evolving problem-solving approach in software industry. However, crowdsourced software engineering is not a risk-free activity, where organizations face various trust issues. To better prepare for such emerging trust issues, this study aims to investigate the critical issues in establishing the trust in context of software crowdsourcing. An industrial survey is conducted to identify the trust issues faced by crowdsourced organizations in conduction of crowdsourced software engineering. The sample of industrial survey comprised of 95 relevant respondents. The authors have identified a list of 11 trust issues. Of these trust issues, 9 have been tagged as critical enclosing 'deficient assistance to best practices', 'malicious code', 'lack of licensed software utilization', 'loss of data', 'network security risks', 'quality of workers', 'social attacks', 'crowd legal action', and 'loss of intellectual property'. The findings of the study are validated via focus group from four academia experts. The results showed that there are no major disagreements between the focus group experts and the industry practitioners. The identified trust issues can eventually permit software development organizations to handle the challenges in development of software in crowdsourced environment and to prepare themselves for any vulnerable situation.

Keywords: trust issues, crowdsourcing, software engineering, industrial survey, empirical study

1. INTRODUCTION

The crowdsourcing is an evolving problem-solving approach grounded on the amalgamation of human and machine working. The word 'crowdsourcing' was first introduced by Howe and Robinson in 2006. Refereeing to the extensively acknowledged definition reported in the article, crowdsourcing is the process, organizations adopt for outsourcing their work to an indeterminate, interacted labor by means of an open call for contribution [1]. Crowdsourced Software Engineering (CSE) originates from the term 'crowdsourcing'. By means of an open call, its employee's worldwide online labor to work on numerous software engineering activities, such as requirements elicitation, design, coding and testing [2]. CSE is executed by numerous prosperous crowdsourcing platforms, such as TopCoder,

Received December 1, 2020; revised January 17 & February 17 & May 24, 2021; accepted June 7, 2021.
Communicated by Chung-Yang Chen.
[+] Corresponding author.

AppStori, uTest, Mob4Hire and TestFlight [3-7]. CSE has also quickly gained attention in industrial and academic communities.

Despite of the fact that it is an emerging paradigm shift of software engineering development that involve crowd over the internet to yields innovative solutions. It is claimed that time to market is reduced in crowdsourcing with the help of parallelism [8-10]. However, without comprehensive trust controlling, a huge number of untrustworthy personnel in the internet crowd can yield to low quality or even scrap answers in the tasks to advantage themselves or damage their opponents' crowdsourcing processes [11].

Loosing trust in both activities of assigning task to crowd and in delivering solutions is critical. Without trustworthy workers in crowdsourcing process, undermines the interests of workers and requesters. According to the study, the trust issues not only significantly enhance the cost of solving any task, but also significantly decrease the usefulness of crowdsourcing processes. Therefore, investigating the trust issues and managing them has become the highest precedence demand in crowdsourcing environments [11]. This is what made us comprehend the need of conducting an industrial survey [12], to create an inventory of trust issues regarding CSE. This leads us to the following research question: What are the critical issues in establishing trust in the context of crowdsourced software engineering (CSE)?

The remainder of this paper is structured as follows. Section 2 introduces the background of research. Section 3 details the methodology employed in this research while Section 4 presents the results of this research. Section 5 reports about categories of identified critical issues. Section 6 reports the results validation through focus group technique used in this study. Section 7 specifies the conclusion, limitations faced in this research, and future work to assist upcoming researcher in this field.

2. BACKGROUND

This section describes background about crowdsourcing, crowdsourced software engineering and the related work on trust issues in crowdsourced software engineering.

2.1 Crowdsourcing

'Crowdsourcing' was primarily accepted in 2006. Jeff Howe introduced the term in his work of 'The Rise of Crowdsourcing', which was later printed in Wired [1]. Latterly in a blog post to this research article [13], the word crowdsourcing was defined as:

"Crowdsourcing represents the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call." Rendering to the above definition, the unclear outsized networked workers and the open call layout are the fundamentals of crowdsourcing. Howe discussed that crowdsourced task can be performed by collaboration or by single individuals [14, 15].

Although the term 'crowdsourcing' has appealed noteworthy attention, the basic notions can be found in numerous former attempts to employee a large appropriately capable labor force in an open call for a particular job in hand [16]. Moving to the modern world of internet, crowdsourcing over the internet can be seen in 2001 [17], when the project

‘InnoCentive’ was sponsored by Eli Lilly to appeal a crowd-based employee outside the organization to help them in development of drug. Likewise, in the similar year, the platform of TopCoder was introduced by Jack Hughes, as a marketplace by means of crowdsourcing for developing software. To simplify the activities performed in distributed software development environment, the TopCoder development technique was recommended [18]. The TopCoder in 2010 has become the largest platform for crowdsourced the software engineering tasks globally.

Thus, claimed advantages of crowdsourcing encompasses easy access to an extensive range of labor over the internet crowd, varied solutions, lesser labor charges and reduced time-to-market. More than 160 projects based on crowdsourcing have been collected with the help of crowdsourcing. It is found that crowdsourcing has been used widely in several disciplines, such as prediction of protein structure [5, 19], retrieval of information [7, 20], discovering drug [4, 21], forecasting the weather [3, 22], for planning transportation [6, 23], and especially in software engineering [8, 24-26].

2.2 Crowdsourced Software Engineering

The broadest sense, the term CSE is used to represent the implementation of techniques regarding crowdsourcing to support the development of software. Various authors have referred this term as ‘Software Crowdsourcing’ or ‘Crowdsourcing Software Development’ or ‘Crowdsourced Software Development’ [27-30]. However, the term CSE is preferred by us as it highlights any of the activity regarding software engineering. These activities include planning, gathering of requirements, improvement in test cases, augmentation of security and others.

Regardless of the extensive usage of crowdsourcing for several software engineering tasks, the notion of CSE is rarely explicitly defined. According to the analysis performed by [31], out of total 210 crowdsourcing based papers surveyed by them, 69% used the notion of crowdsourcing without quoting its definition. According to the authors, 18% of the studies cited the definition and it was from Howe’s work. Only two of the studies explicitly reported the crowdsourcing with the perspective of software engineering activities [8, 32].

The definition by Stol and Fitzgerald’s [8] extends Howe’s definition of crowdsourcing to the paradigm of software engineering, demanding the indeterminate work force to have essential expert knowledge. Likewise, the definition from Huhns [32] is dignified as a Wikipedia page on the topic of software crowdsourcing. Since Howe’s crowdsourcing definition is widely accepted, so it is chosen to be used for this study as well. The formal definition of CSE is “Crowdsourced software engineering is the act of undertaking any external software engineering tasks by an undefined, potentially large group of online workers in an open call format”.

2.3 Trust Issues Relevance in Crowdsourced Software Engineering

In the past few decades, trust has been extensively debated across various disciplines including economics, psychology and computer science [33]. In broader sense, trust is enlightened as a relationship among a trustier and a trustee, which specifies that the trustier have faith in trustee within a precise context [34]. It is significant to model trust as it can support the collaboration among multiple entities move on the way to an improved result than the estimated one. However, modeling trust is a complex issue because numerous factors may together affect the trust among two entities [11].

In context of software crowdsourcing, various issues are emerging. These issues can be regarding communication and coordination, data security, customer enrollment and the most troublesome issue is regarding trust. During task selection, allocating task to crowd and getting the timely response with high quality results involves trust. It is claimed that crowdsourcing works in parallelism [8-10] that helps to reduce the time to perform the task. However, it is equally important to have trust control in order to have trustworthy personnel for high quality solutions to the allocated tasks [11]. According to the recent study conducted by Wang in 2019 [11], it is important to investigate the trust issues in crowdsourcing platform as the trust issues not only enhances the cost of solving any task but also decreases the crowdsourcing usefulness.

Although few studies have highlighted the importance of trust issue in crowdsourcing setting [3, 6, 8, 11, 16]. However, there is less research found in context of investigating what are the critical issues in establishing much needed trust in the context of crowdsourcing to perform software engineering tasks. Why trust is still challenge for tasks distribution, selection of crowd, time to market and others. Ignoring the fact of establishing trust in such dynamic crowdsourcing setting, it can damage its spirit by not having innovative solutions and may underutilize the brains of internet crowd. Therefore, investigating the trust issues has become the highest precedence demand in crowdsourcing environments [11].

3. METHODOLOGY

A quantitative methodology is adapted in this study to investigate the critical issues in establishing the trust in context of software crowdsourcing. Thus, after careful review of the literature, the research question of this study is answered through conducting an industrial survey through questionnaire among the professionals of software crowdsourcing industry. Among various research methods, the authors have chosen survey method due to its comprehensiveness and advantages specified in literature [35]. The advantages include collection of data from large number of respondents irrespective of their geographic locations, cost and time effectiveness, convenient for the software engineer to fill the questionnaire whenever has time, and many others [36]. There are various other studies [37-39], who have also used the same methodological approach in this field. After the results collected from survey, these trust issues are categorized into critical and non-critical issues. These resulted identified issues are then validated by using the focus group technique.

The authors of this study designed the questionnaire regarding critical issues establishing trust in crowdsourced software engineering. The questionnaire was developed by following the Mark Kasunic guideline [12]. The researchers used single type of questionnaire format. The authors chose an open-ended questionnaire format as an instrument to gather data. The questions are shown in Appendix A. The instrument is validated for face validity and content validity through two methods; Average Congruency Percentage (ACP), and Content Validity Index (CVI) Researchers when deal with any newly developed scale, they need to provide its reliability and validity. The content validity of the scale is measured as significant in generating conclusions regarding the quality of the scale. The content validity relates with the degree to which some items together, make an acceptable working definition of a construct.

In this research, CVI for the individual item is identified (I-CVI). There is a general agreement related to compute item-level CVI (I-CVI). In this a group of content experts is requested to rate each item of the scale for its relevance to the mentioned constructs.

A minimum of three experts are required to conduct this validation process. In this validation process, the item ratings are usually on a 4-point ordinal scale. For each item, I-CVI is calculated based on the ratings the experts have given divided by the total number of experts. For example, if an item is rated as highly relevant by three out of four experts would have an I-CVI of .80.

Piloting of the questionnaire was done through fellow researchers, and their suggestions were incorporated accordingly. Moreover, the instrument was also validated through four experts. It was also ensured that experts must have educational and research background in crowdsourcing and software engineering. All the experts were selected who have more than 10 years of software development experience.

As shown in Table 1, for ACP experts computed the percentage of questions deemed to be relevant for them. Whereas in CVI, the content validity index for individual item (I-CVI) was calculated. The experts rated questions for their relevancy. Expert 3 and expert 2 found one out of 8 questions irrelevant, resulting 91.66% relevancy at their individual level. Whereas expert 2 and expert 1 rated all questions relevant and resulted to 100% relevancy rate at their individual level. The average value of the experts' congruency percentage is 95.5%, which is considered valid. For CVI, panel of same four experts were asked to evaluate each question's content relevancy on 4-point likert scale. Where 1 = not relevant, 2 = somewhat relevant, 3 = relevant, 4 = very relevant. For each of the question, to decide the criteria for relevancy, the number of experts giving 3 or 4 score was counted as relevant, and 1 or 2 was considered as not relevant. Table 1 shows the CVI results. Each of the expert responses is notated with '×'. It notates the agreement of the expert towards the relatedness of any question.

On basis of the results (ACP, I-CVI), the face and content validity of questions to be asked in questionnaire were found significantly high, hence ensuring the quality of the instrument. The authors distributed questionnaire across the target audience in two distinct ways. These included the online version, and a hard copy. The online survey was developed using Google survey tool. The details of the steps followed in survey are reported below.

Table 1. Details of the participated experts for questionnaire validation.

Questions	Expert 1	Expert 2	Expert 3	Expert 4	Number of Agreement	I-CVI
1	–	×	×	×	3	0.75
2	×	×	×	×	4	1.00
3	×	×	–	×	3	0.75
4	×	–	×	×	3	0.75
5	×	×	×	×	4	1.00
6	–	×	×	×	3	0.75
7	×	×	×	×	4	1.00
8	×	×	×	×	4	1.00
Proportion Relevant	0.75	0.87	0.87	1.00	Mean I-CVI	0.87
	Mean Expert Proportion = 0.87					

3.1 Data Source

The survey comprised of the following main tasks. At first, the stakeholders were identified. For the identification of the most relevant stakeholders, the survey invitation was sent to various software development companies. A total of 12 companies were contacted and 9 of them showed their willingness to respond to the invitation. The selected software development companies are shown in Table 2. In total 118 relevant respondents (employees of the selected companies) showed their willingness to participate in the survey. Consequently, the authors sent the questionnaire through web link as well as distributed the hard copies to some of the respondents. However, the authors managed to receive 98 responses.

Once the responses were gathered, the validity of the responses was also checked. Latterly, the filtration was done on the gathered responses and 3 responses (questionnaires) were further discarded leaving behind 95 responses in total of a response rate 80%. The detail response rate of the respondents is shown in Fig. 1.

As shown in Fig. 1, the vertical bar shows the number of responses and the horizontal bar shows the months of responses. The respondents participated in the survey belong from different countries including UK, USA, Canada, Malaysia, Pakistan and Australia. The survey call was sent to the survey call was sent for all experience range (junior to senior) but the selected responses from participants (complete in information) ranged from 4.5 to 16 years of crowdsourcing experience. The graph in fig 1 shows the distribution of number of responses gathered in the various months. 10 responses were gathered in November, 5 in December, 25 in January, 2 in February, 12 in March, 15 in April, 6 in May, 11 in June, 4 in July, and 5 in August. This ultimately results in 95 total responses.

Table 2. Detail of software development companies.

Sr. No.	Name of Software Companies	I-CVI
1	Rainforest	https://www.rainforestqa.com
2	Kaggle	https://www.kaggle.com
3	TopCoder	https://www.topcoder.com
4	OneByte	https://onebytellc.com
5	Ninesol Technologies	http://www.ninesol.com
6	Fortlogics Solutions	https://fortlogics.com
7	Crowdsourced testing	https://crowdsourcedtesting.com/en
8	Global App Testing	www.globalapptesting.com
9	Crowdsprint	https://crowdsprint.com

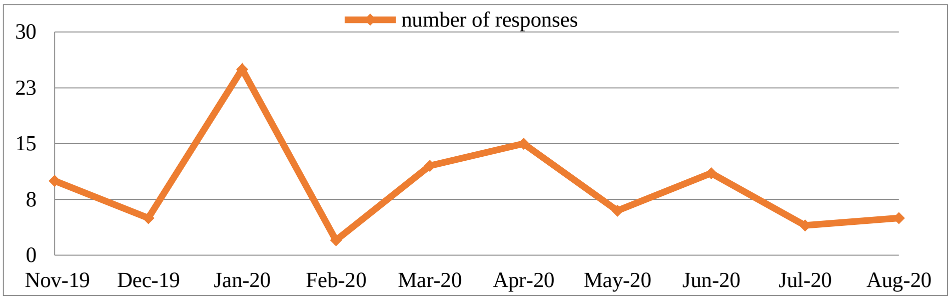

Fig. 1. Respondents response rate.

3.2 Data Analysis

The critical issues establishing trust regarding crowdsourced software development are identified. Frequency analysis technique is performed to analyze these trust issues. The frequencies technique is chosen by the authors as it is one of the appropriate methods to analyze the qualitative data [40]. Thus, the authors have analyzed the occurrences of the trust issues in this study. For that, frequency is measured for each trust issue, as shown in Table 3. Each trust issue is analyzed by calculating its presence in the filled questionnaire. Furthermore, the comparative significance of each trust issue is also identified by associating the presence of one trust issue against another trust issue. The details are reported in the subsequent sections.

4. RESULTS

In this section, the results are presented and discussed in detail. The study identified a total of 11 trust issues regarding crowdsourced software engineering. The detail of each trust issues is described in the following subsections.

4.1 Trust Issues Identified via Survey

By responding research question of this study (see Section 1), the authors came up with a list of 11 issues in establishing trust through survey via questionnaire as shown in Table 3. Of the identified trust, issues, 9 were marked as critical trust issues based on 30% frequency criteria, *i.e.* a trust issue with a frequency ≥ 30% has been marked as a critical trust issue. This 30% frequency criteria is widely used in literature for investigating the critical factors [39]. Thus, the study in hand has also used this criteria for investigating the critical trust issues in context of crowdsourced software engineering.

Table 3. List of identified trust issues through survey.

Sr. No.	Trust Issues	I-CVI
1	Deficient assistance to best practices	91
2	Malicious code	85
3	Lack of licensed software utilization	83
4	Loss of data	81
5	Network security risks	79
6	Quality of workers	76
7	Social attacks	73
8	Crowd legal action	67
9	Loss of intellectual property	61
10	Overall worker reputation	26
11	Verification of data	20

A graphical representation of identified critical issues establishing trust is depicted in Fig. 2. The critical issues are represented in terms of frequencies. It is notable that highest frequency of 95% is found for ‘Deficient assistance to best practices.’ Whereas, ‘Malicious code’ (89%), ‘Lack of licensed software utilization’ (87%), ‘Loss of data’ (85%), ‘Network security risks’ (83%), ‘Quality of Workers’ (80%), ‘Social attacks’ (77%), ‘Crowd Legal Action’ (70%), and ‘Loss of Intellectual property’ (68%) yield above rate of 60. Only two trust issues named 'Overall Workers Reputation' and 'Verification of Data' are found with the score of 26% and 20%. The reason of getting low count to these two issues is obvious as industry practitioners will not rate their reputation as trust issue. The detail description of each of the identified trust issues is explained in the following section.

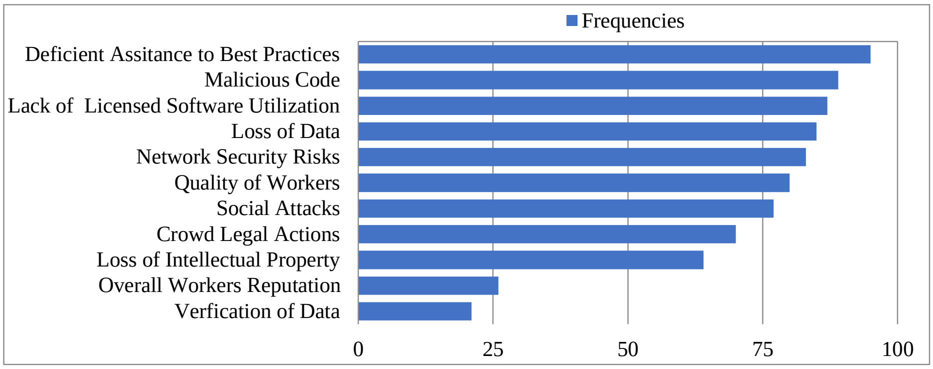

Fig. 2. Identified critical issues in establishing trust response rate.

4.2 Deficient Assistance to Best Practices

Table 3 specifies that 'Deficient Assistance to Best Practices' had the highest frequency (95%), so it is ranked highest in our findings. According to the respondents, in software development context, security of the software is matter of technology and methods combined alongside people involved in the process. It is argued that in crowdsourced software development, massive numbers of people are involved adding a possible risk of security. Uniformity in standardizing and assisting others to follow is still scarce. As respondents had emphasized on using standards for implementation of tasks to ensure product or service security. They further mentioned the necessity of the business policy establishment, its awareness and understanding the conformity to numerous privacy regulations associated in developing software in crowdsourced environment.

4.3 Malicious Code

'Malicious code' had the second highest frequency (89%) in our findings (as shown in Table 3). The respondents specified that the malicious code (harmful) is always a concern in crowdsourcing. Due to this, there is always a chance to submit more than the required functionality. According to the respondents, the crowd can easily include malicious code in the program, that later can be used for exploitation, resulting in numerous security breaches. Some of the respondents considered this act of inserting malicious code as carelessness and some has considered it as malicious intention of the crowd.

4.4 Lack of Licensed Software Utilization

Software Utilization without the license is reported as third most cited trust issue (87%) by the respondents. Respondents reported that in correspondence with crowd trustworthiness, it is important to first understand the compliance requirements before understanding any business. It helps to validate the submissions for adhering to compliance regulations. According to the respondents, it happens that the individual in the crowd copies the solution from the tools and API by using any third party, where it was not an open source code library. Such way to use licensed software is considered as a violation of compliance regulations.

It was also argued by the respondents that ultimately this can lead to a major impact. Moreover, some respondents reported another scenario where the non-compliant licensed software is used. In this scenario the individual in the crowd can resubmit the developed application at smoother place, which again violates the compliance regulations.

4.5 Loss of Data

'Loss of Data' is the fourth highly cited trust factor (85%) in the findings. According to the respondents one of the domains where risks regarding crowdsourcing usually run forth of law is data security. Crowdsourcing is picked in various cases to ease and encourage research by holding the crowd's originality but at the cost of sharing information that might be sensitive. According to the respondents, there is always a chance of losing the sensitive information when people in the crowd share the data, leading to numerous privacy and security breaches.

4.6 Network Security Risk

'Network Security Risk' is reported by 83% of the respondents (as shown in Table 3). According to the respondents, it is necessary for the organizations to give network access to the crowd involved in software development. Due to such access, the sensitive information usually gets exposed. The respondents further reported that, such access is usually not monitored carefully, that ultimately results in serious concerns towards confidentiality.

4.7 Quality of Workers

'Quality of Workers' is cited by 80% of the respondents. According to them, there has to be some mechanism that can evaluate the worker quality in crowdsourcing. It is necessary to ensure the authenticity of their provided answers. Some other respondents mentioned the need of a model or a technique that can help to identify the spam and biased workers. Similarly, it is also mentioned by the respondents to evaluate the reliability of the worker by using some technique. Respondents further mentioned that not measuring the worker quality often leads to low quality solutions or wrong answers.

4.8 Social Attacks

The trust issue 'social attacks' is reported by 77% of the respondents. According to them it is one of the most commonly occurring issue in which individual in the crowd after agreeing on any given schedule to complete a task backs off. Some of the respondents mentioned the same issue by reported another scenario in which people in the crowd, either having competitive or malicious intention, refuse to submit the task on the consented deadlines. It is further elaborated by the respondents that due to such scenarios, organizations need to re-initiate the complete process resulting extra burden on the organization in terms of the resources (time, cost, and effort).

4.9 Crowd Legal Actions

In our findings, ‘Crowd Legal Action’ is also identified as another critical trust issue with a frequency of 70%. The respondents of the survey reported about the confidential innovated information that is possessed by almost every solution. According to the respondents, it usually happens when the individual in the crowd who is non-winning entity charge the client for taking and integrating his/her idea in the business process. Respondents further shared their view about the federal and state laws regarding the relationship that has to be maintained among the employer and the employee. They mentioned the lack of clear definition of such laws over employment practice in crowdsourcing. As a result, often the employer is accused by the crowd based on the provided benefits.

4.10 Loss of Intellectual Properties

The authors also identified ‘Loss of intellectual property’ as another critical trust issue with a frequency of 68% as shown in Table 3. The respondents of the survey reported about the intellectual property risks that often emerge when a company adopts a crowdsourcing technique to develop software. According to the respondents, the task at first is provided in modular form. It is always very difficult to make the crowd understand about the problem completely. The modules are always designed in simplified way, so that they don't give big picture to the crowd. Such reluctance to share details of the tasks leads to less understanding of the requirements. To maintain a balance between them is a challenge and a risk towards crowd trustworthiness. According to the respondents of the survey, the organizations often lose their competitive advantage due to loss of their intellectual property. It happens as there is no clarity about owning the solution – the crowdsourcing initiation organization or the individual in the crowd who has submitted the solution.

5. CATEGORIES OF IDENTIFIED ISSUES AS CRITICAL AND NON-CRITICAL

Although the above identified critical issues in establishing trust are covers different perspectives. However, authors of this study have categorized them into four facets according to their similar nature and purpose. These four categories are data, legality, security and workers. The Data category is crucial and is baseline for any further correspondence for software development. And the trust issue emerging because of data is critical and industry must prepare themselves strongly to avoid it. Legality in software crowdsourcing is another emerging aspect that need to be resolved by appropriately establishing the policies and protocols of its usage. Providing security over software crowdsourcing is another important aspect that industry must look towards to avoid any attacks, risks and vulnerabilities. Workers category is the core to establish trust, and if it is ignored can impact very strongly to the overall quality of software and can also destabilize the whole software crowdsourcing process. Moreover, the reason of establishing these categories is to assist industry to better prepare in all these crucial aspects. They should be advancing their efforts towards these categories, rather should deploy some dedicated professional to avoid any such emerging trust issues. Fig. 3 shows the association of 11 identified critical trust issues into four categories.

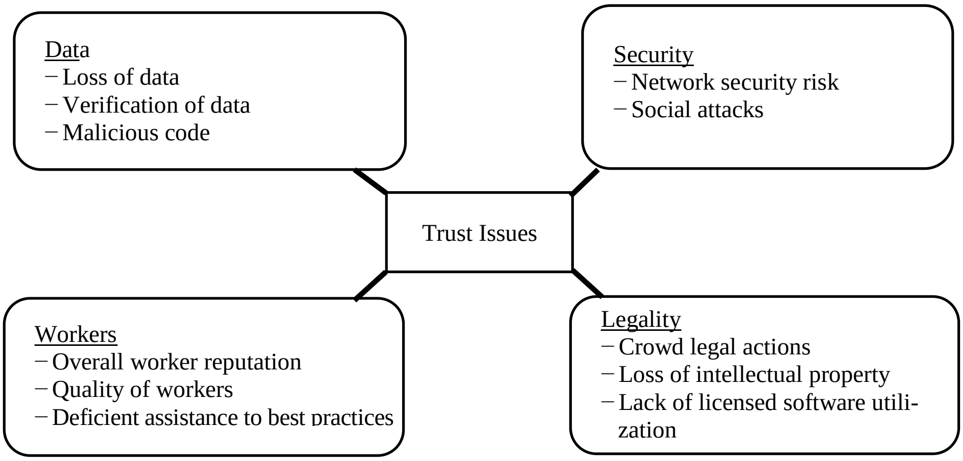

Fig. 3. Categories of identified critical issues in establishing trust.

These categories names are given based on the similar nature and functional association of issues who are taking role of in establishing trust in software crowdsourcing. In this regard, the 'Data' category is given to the three issues related to the data of loss of data, verification of data and malicious code. Similarly, 'Legality' category named is selected against the issues of crowd legal actions, loss of intellectual property and lack of licensed software utilization. Moreover, 'Security' category is made to refer the network security risk, social attacks. Lastly, 'Workers' category is given to cover most of the issues of overall worker reputation, quality of workers and deficient assistance to best practices.

This categorization is made to highlight the abstract level of alignment of these identified critical trust issues. It helps to understand that what four abstract categories are considered imperative that reflects the concerns in establishing trust in software crowdsourcing.

6. VALIDATING RESULTS AND SUGGESTING SOLUTION STRATEGIES THROUGH FOCUS GROUP TECHNIQUE

The focus group is conducted to validate the findings of this research (Trust Issues regarding crowdsourced software engineering). Besides the solution strategies are also suggested with the help of focus group discussions. Focus groups are the discussions which are cautiously planned. Focus group typically comprises of 3 to 12 participants where the moderator guides and facilitates the discussion. The moderator follows an outlined structure in order to keep the discussion focused. The focus group members are selected on basis of their characteristics relevant to area of discussion. This is called purposive sampling. The settings of the group give the authority to the participants to set up the responses and ideas, ultimately increases the depth of the gathered information [41]. Focus group sessions mainly generate qualitative knowledge about the study. There are various benefits of focus group. Like focus group generate candid, intuitive information. Besides focus group is found to be inexpensive to be performed [42]. Currently, there are various studies who used this method, *e.g.*, in market research [43], system usability studies [42], product planning [43], and business services [44]. Various guidelines are available to conduct the focus group in an effective manner [41, 45], producing a method that is comparatively easy to be embraced and used.

It was a comprehensive process that consisted of six steps. At first the objective for focus group conduction was defined. The objective to conduct the focus group for this research is to validate the list of trust issues regarding crowdsourced software engineering. Secondly, the fixed time line of the focus group was established. In this case the timeline to conduct the focus group and validate the outcome was three months.

The planning phase of the focus group session initiated six weeks prior to the main session. By viewing the profiles of the software engineers, seven senior software engineers who have at least 10 years of crowdsourcing experience as a project manager, were contacted. Six among them showed their consent that followed by the formal invitations through email. These experts belonged from the companies shown in Table 2. The experts of the focus group did not participate in survey (interview session). It was assured that all these experts must have been working in crowdsourcing environment. The list of task details was also shared with the experts (comprising of the task to be performed and the detail about the focus group session). The questions that needed to be asked from experts were prepared thoroughly. The sample questions are shown in Table 4.

As shown in Table 4, the provided sample questions were asked in the focus group's main session. A senior researcher of the study acted as a facilitator, who opened the session. At first the participants were welcomed by the facilitator. Besides, the reason of the focus group was reviewed together with the objective of the meeting. Facilitator made sure that the whole thing goes over the stream of the meeting. By doing this, the authors of the research laid out the basic rules.

The focus group session was initiated with an open question. It was a general question ("What do you think about Crowdsourcing based Software Engineering?"). The authors of the research made sure that entire views on numerous questions got a chance to be picked up carefully. The comments of the focus group experts on the identified trust issues regarding crowdsourced software engineering were recorded.

Table 4. Sample of open ended questions for focus group.

No.	Sample of open ended questions
1	What do you think about crowdsourced based software engineering?
2	Do you think that proposed study has comprehensively covered the various trust issues regarding crowdsourced software engineering?
3	What are the strengths and weaknesses of the research outcome (trust issues regarding crowdsourced software engineering)?
4	What aspects of the study can be improved and how?

6.1 Comprehensiveness of Outcome

All the experts had consensus on the comprehensiveness of the identified trust issues. According there is substantial variability in outcome reporting the trust issues that can help the practitioners working in domain of crowdsourced software engineering. According to expert 6 and 3, although the study has identified 11 trust issues, but the comprehensive survey can help in generalizing the results.

6.2 Strength of the Study

According to the focus group experts, the outcome of the study is easy to comprehend and understand. The experts reported about the realism of the identified trust issues by mentioning their frequency analysis where majority of the trust issues are having more than 30% of the frequency, showing them most commonly occurring trust issues. Besides the experts in the focus group mentioned a need to consult the state of the knowledge and then to correlate the findings. The comments of the expert are noted and mentioned as a future work.

6.3 Aspects of the Study to Be Improved

Although the focus group experts considered the findings of this study as comprehensive and valuable contribution towards the crowdsourced software engineering body of knowledge. However, according to the experts, this study can be further improved if the existing literature would also be consulted. Furthermore, it is mentioned by focus group that correlation between the findings of the literature and industry would be another aspect towards the improvement of results.

6.4 Solution Strategies to Overcome Trust Issues

The focus group suggested some solution strategies for overcoming trust issues in crowdsourced environment. According to the members of the focus group, for having practices into action, there is a need to first develop the implementation strategies. They further mentioned the necessity of the business policy establishment, its awareness, and understanding the conformity to numerous privacy regulations associated with developing software in crowdsourced environment.

Furthermore, the focus group discussed the issue of malicious code. They came with the suggestion that the acceptability of any response should be analyzed on basis of the extent to which prior reported expectations are met. According to them it is important to accurately interpret the responses of the worker for measuring the code maliciousness in a crowdsourced environment. For overcoming the issue of lack of licensed software utilization, the members of the focus group reported that it is significant to understand the adherence to requirements before understanding any business. It lends a helping hand to validate the submissions for adhering to compliance regulations.

Similarly, the focus group discussed the issue of ‘loss of data’. They came up with the suggestion that in crowdsourced environment, data transmission should be done with the help of appropriate mediums like smart phones via internet or via mobile phone networks. Furthermore, the supervisory control and data acquisition can also be a better solution for avoiding data loss in its transmission.

Network security risk is another issue that is faced in crowdsourced environment. The focus group suggested that the crowdsourcing organizations must have understand the cybersecurity concerns with that of the surrounding issues related to variety of cyberattack. Besides they should also know how to devise the countermeasures. The members of the focus group further mentioned if such understandings would be developed, it will help in preserving the confidentiality, and upstandingness.

Likewise, the issue of 'quality workers' is discussed in focus group for its overcoming solution strategies. According to the focus group members, any approach should be used to calculate and investigate the expertise of workers in a crowdsourced environment. It would be better if the chosen approach would be based on the workers score modelling by using graphs as it will help in better understanding and analysis.

The issue of 'social attacks' was discussed in focus group session. The members suggested for an approach that can detect social and cyber-attacks. The discussion in focus group concluded with the suggestion that the detection approach should work on event triggers, where less or no training is required with labeled samples.

Furthermore, the member of the focus group discussed the solution strategy for the issues 'crowd legal action' and 'Loss of Intellectual Properties'. According to them there should be clear copyright laws that should be disseminated among the crowd and the and the crowdsources. Besides they mentioned the need of license contracts. According to them it is essential for the crowdsources to have such license contracts clearly defining the projects contributors in crowdsourcing environment.

7. CONCLUSION, LIMITATION AND FUTURE WORK

Crowdsourced software engineering has become a mutual aim across the software industry in the globe. However, without comprehensive trust controlling, a huge number of untrustworthy personnel can yield to low quality work that subsequently affect the software industry. According to the study, the emerging trust issues not only significantly enhance the cost of solving any task, but also significantly decrease the usefulness of crowdsourcing processes. Therefore, authors have addressed this emerging concern by identifying the trust issues regarding crowdsourced software engineering. It can permit software development organizations to handle the trust related challenges in the development of a software in crowdsourced environment.

In this study, an industrial survey is conducted among the professionals of software crowdsourcing industry via questionnaire survey. Authors have identified a list of 11 trust issues. Of these issues, 9 have been tagged as critical trust issues. These critical trust issues are 'deficient assistance to best practices', 'malicious code', 'lack of licensed software utilization', 'loss of data', 'network security risks', 'quality of workers', 'social attacks', 'crowd legal action', and 'loss of intellectual property'. The findings of the study are validated via focus group from 4 experts. The validation results show that there are no major disagreements between the focus group experts and the industry practitioners.

This study is not without limitations. One of the limitations is subject to survey response as a total of 95 relevant responses acted as a final sample with a response rate of

80%. One of the difficulties with the survey is very low response rate besides with the possibility of subjective partialities. It is reported in literature that the responses gathered via survey may have difference with real population distribution besides with biasness [45]. However, the authors have tried their level best to discover the practitioners' experiences regarding crowdsourcing but their experiences are not verifiable. Another limitation is also possible that the authors have got the inaccurate experiences of the practitioners' perceptions. The representative sample for the survey is 95 and majority of them were foreign experts, addressing the external validity of the study. One of the limitations of this study is relevant to the experts for results validation that can be increased later on for further verifying the results.

In future research, researchers can work on exploring this emerging paradigm in lens of different software development activities. Study in hand has identified trust issues without considering any specific activity, but upcoming research can be more useful if researchers focus on target a particular development activity like documentation, coding or testing and explore the issues and possible solution to these issues. Furthermore, in future the possible theoretical or practical treatments/implication (solutions of trust issues) from the state of knowledge and practice can be explored and identified. Another stream of future research can be related to the finding of this study. As this study has explored and categorized the identified issues as critical and non-critical issues. However, future research can develop and use any model to validate and conform the findings (issues as critical and non-critical) of this study. Besides, the authors are dedicated to the subsequent work in the future such as conduction of systematic literature review to identify the trust issues regarding crowdsourced software engineering reported in previous studies, and correlating the findings with the industry-based findings. Moreover, authors are aiming to develop a software tool to assist crowdsourced organizations to have trustworthy crowd to perform the generated tasks.

REFERENCES

1. J. Howe, "The rise of crowdsourcing," *WIRED*, Vol. 14, 2006, pp. 1-4.
2. M. N. Malik and H. H. Khan, "Investigating software standards: A lens of sustainability for software crowdsourcing," *IEEE Access*, Vol. 6, 2018, pp. 5139-5150.
3. T. D. LaToza and A. van der Hoek, "Crowdsourcing in software engineering: Models, motivations, and challenges," *IEEE Software*, Vol. 33, 2015, pp. 74-80.
4. I. Christensen and C. Karlsson, "Open innovation and the effects of crowdsourcing in a pharma ecosystem," *Journal of Innovation & Knowledge*, Vol. 4, 2019, pp. 240-247.
5. Y. A. Alotaibi, "New secured e-government efficiency model for sustainable services provision," *Journal of Information Security and Cybercrimes Research*, Vol. 3, 2020, pp. 75-96.
6. M. Shergadwala, H. Forbes, D. Schaefer, and J. H. Panchal, "Challenges and research directions in crowdsourcing for engineering design: An interview study with industry professionals," *IEEE Transactions on Engineering Management*, Vol. 1, 2020, pp. 1-13.
7. E. Maddalena, S. Mizzaro, F. Scholer, and A. Turpin, "On crowdsourcing relevance magnitudes for information retrieval evaluation," *ACM Transactions on Information Systems*, Vol. 35, 2017, pp. 1-32.

8. K. J. Stol, B. Caglayan, and B. Fitzgerald, “Competition-based crowdsourcing software development: A multi-method study from a customer perspective,” *IEEE Transactions on Software Engineering*, Vol. 45, 2017, pp. 237-260.
9. S. Shafiq and I. Inayat, “Model-driven development based cross-platform development: A review,” *Journal of Information Science and Engineering*, Vol. 33, 2017, pp. 1561-1573.
10. O. I. Khalaf, M. Sokiyna, Y. Alotaibi, A. Alsufyani, and S. Alghamdi, “Web attack detection using the input validation method: dpda theory,” *Computers, Materials & Continua*, Vol. 68, 2021, pp. 3167-3184.
11. Y. Wang, Y. Lin, Z. Gao, and Y. Chen, “A two-stage iterative approach to improve crowdsourcing-based relevance assessment,” *Arabian Journal for Science and Engineering*, Vol. 44, 2019, pp. 3155-3172.
12. M. Kasunic, *Designing an Effective Survey*, Software Engineering Institute, Carnegie Mellon University, 2005.
13. J. Howe, “Crowdsourcing: A definition,” *Typepad*, 2006.
14. F. Rodrigues, M. Lourenco, B. Ribeiro, and F. C. Pereira, “Learning supervised topic models for classification and regression from crowds,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, 2017, pp. 2409-2422.
15. L. M. Maruping, S. L. Daniel, and M. Cataldo, “Developer centrality and the impact of value congruence and incongruence on commitment and code contribution activity in open source software communities,” *MIS Quarterly*, Vol. 43, 2019, pp. 951-976.
16. X. Zhang, L. Shangguan, and Y. Yuan, “A crowd wisdom management framework for crowdsourcing systems,” *IEEE Access*, Vol. 4, 2016, pp. 9764-9774.
17. K. Mao, L. Capra, M. Harman, and Y. Jia, “A survey of the use of crowdsourcing in software engineering,” *Journal of Systems and Software*, Vol. 126, 2017, pp. 57-84.
18. J. M. Hughes *et al.*, “System and method for software development,” United States Patent Application 20060184928, Kind Code: A1.
19. Y. Han, P. Ozturk, and J. V. Nickerson, “Leveraging the wisdom of the crowd to address societal challenges: Revisiting the knowledge reuse for innovation process through analytics,” *Journal of the Association for Information Systems*, Vol. 21, 2020, p. 8.
20. J. Love and R. Hirschheim, “Crowdsourcing of information systems research,” *European Journal of Information Systems*, Vol. 26, 2017, pp. 315-332.
21. D. C. Thompson and J. Bentzien, “Crowdsourcing and open innovation in drug discovery: recent contributions and future directions,” *Drug Discovery Today*, 2020, pp. 2284-2293.
22. Y. Alotaibi and F. Liu, “How to model a secure information system (IS): A case study,” *International Journal of Information and Education Technology*, Vol. 2, 2012, p. 94.
23. X. Wan, H. Ghazzai, and Y. Massoud, “Mobile crowdsourcing for intelligent transportation systems: Real-time navigation in urban areas,” *IEEE Access*, Vol. 7, 2019, pp. 136995-137009.
24. Y. Alotaibi, M. N. Malik, H. H. Khan, A. Batool, S. U. Islam, A. Alsufyani, and S. Alghamdi, “Suggestion mining from opinionated text of big social media data,” *Computers, Materials & Continua*, Vol. 68, 2021, pp. 3323-3338.

25. C. Li, L. Huang, J. Ge, B. Luo, and V. Ng, "Automatically classifying user requests in crowdsourcing requirements engineering," *Journal of Systems and Software*, Vol. 138, 2018, pp. 108-123.
26. Y. Alotaibi, "A new database intrusion detection approach based on hybrid meta-heuristics," *CMC-Computers Materials & Continua*, Vol. 66, 2021, pp. 1879-1895.
27. Z. Hu, W. Wu, J. Luo, X. Wang, and B. Li, "Quality assessment in competition-based software crowdsourcing," *Frontiers of Computer Science*, Vol. 14, 2020, pp. 1-14.
28. D. Firmenich, S. Firmenich, J. M. Rivero, and L. Antonelli, and G. Rossi, "Crowd-Mock: an approach for defining and evolving web augmentation requirements," *Requirements Engineering*, Vol. 23, 2018, pp. 33-61.
29. S. Bibi, I. Zozas, A. Ampatzoglou, P. G. Sarigiannidis, G. Kalampokis, and I. Stamelos, "Crowdsourcing in software development: Empirical support for configuring contests," *IEEE Access*, Vol. 8, 2020, pp. 58094-58117.
30. D. Zahay, N. Hajli, and D. Sihi, "Managerial perspectives on crowdsourcing in the new product development process," *Industrial Marketing Management*, Vol. 71, 2018, pp. 41-53.
31. A. Sari, A. Tosun, and G. I. Alptekin, "A systematic literature review on crowdsourcing in software engineering," *Journal of Systems and Software*, Vol. 153, 2019, pp. 200-219.
32. M. N. Huhns, W. Li, and W. T. Tsai, "Cloud-based software crowdsourcing (dagstuhl seminar 13362)," *Dagstuhl Reports*, Vol. 3, 2013, No. 9.
33. S. Hantke, T. Olenyi, C. Hausner, T. Appel, and B. Schuller, "Large-scale data collection and analysis via a gamified intelligent crowdsourcing platform," *International Journal of Automation and Computing*, Vol. 16, 2019, pp. 427-436.
34. Q. Zhang, D. DiFranzo, M. J. K. Gloria, B. Makni, and J. A. Hendler, "Analyzing the flow of trust in the virtual world with semantic web technologies," *IEEE Transactions on Computational Social Systems*, Vol. 5, 2018, pp. 807-815.
35. Y. Alotaibi, "Automated business process modelling for analyzing sustainable system requirements engineering," in *Proceedings of IEEE 6th International Conference on Information Management*, 2020, pp. 157-161.
36. M. Zahedi, M. Shahin, and M. A. Babar, "A systematic review of knowledge sharing challenges and practices in global software development," *International Journal of Information Management*, Vol. 36, 2016, pp. 995-1019.
37. A. A. Khan, J. Keung, M. Niazi, S. Hussain, and A. Ahmad, "Systematic literature review and empirical investigation of barriers to process improvement in global software development: Client-vendor perspective," *Information and Software Technology*, Vol. 87, 2017, pp. 180-205.
38. M. Ilyas and S. U. Khan, "Software integration in global software development: challenges for GSD vendors," *Journal of Software: Evolution and Process*, Vol. 29, 2017, p. e1875.
39. F. Palomba, M. Linares-Vásquez, G. Bavota, R. Oliveto, M. di Penta, D. Poshyvanyk, and A. de Lucia, "Crowdsourcing user reviews to support the evolution of mobile apps," *Journal of Systems and Software*, Vol. 137, 2018, pp. 143-162.
40. M. Salam and S. U. Khan, "Challenges in the development of green and sustainable software for software multisourcing vendors: Findings from a systematic literature

review and industrial survey," *Journal of Software: Evolution and Process*, Vol. 30, 2018, p. e1939.

41. J. Langford and D. McDonaugh, *Focus Groups: Supporting Effective Product Development*, Taylor and Francis, London, 2003,
42. R. Widdows, T. A. Hensler, and M. H. Wyncott, "The focus group interview: A method for assessing user's evaluation of library service," *College and Research Libraries*, 1991, pp. 352-359.
43. J. Rubin, *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*, John Wiley & Sons, NJ, 1994.
44. S. L. Baker, "Improving business services through the use of focus groups," *Reference Quarterly*, Vol. 30, 1991, pp. 377-385.
45. N. Abbas, J. Andersson, and D. Weyns, "ASPLe: A methodology to develop self-adaptive software systems with systematic reuse," *Journal of Systems and Software*, Vol. 167, 2020, p. 110626.

APPENDIX A: SURVEY QUESTIONS

Q1. What are the trust issues you have faced while indulged in a crowd sourced software development?

Q2. Do you face any data related trust issue that affects the software development process in a crowdsourced environment? If yes, what are they?

Q3. Do you face any security related trust issue that affects the software development process in a crowdsourced environment? If yes, what are they?

Q4. Do you face any legality related trust issue that affects the software development process in a crowdsourced environment? If yes, what are they?

Q5. Do you face any workers related trust issue that affects the software development process in a crowdsourced environment? If yes, what are they?

Q6. Do you think over all workers reputation is a trust related issue that affects the software development process in crowdsourced environment?

Q7. Do you think network security and social attacks are trust issues that can affect the software development process in crowdsourced environment?

Q8. Do you think malicious code is a trust related issue that can affect software development process in crowdsourced environment?

Huma Hayat Khan obtained her PhD in Software Engineering in the Advanced School of Informatics at the Universiti Teknologi Malaysia, Malaysia. Currently, she is Assistant Professor in the Faculty of Engineering and Computer Science at the National University of Modern Languages, Islamabad, Pakistan. Her main research interests are focused on 18 software requirement engineering, situational software engineering, global software development, block chain and software outsourcing.

Muhammad Noman Malik obtained his PhD in Computer Science from Razak School of Engineering and Advance Technology at the Universiti Teknologi Malaysia, Malaysia. Currently, he is Assistant Professor in the Faculty of Engineering and Computer Science at the National University of Modern Languages, Islamabad, Pakistan. His main research interests are focused on human computer interaction, software defect prediction and human values in software engineering.

Youseef Alotaibi is an Associate Professor in the Department of Computer Science, College of Computer and Information Systems, at Umm Al-Qura University, Saudi Arabia. He completed his PhD from the Department of Computer Science and Computer Engineering, La Trobe University in, Melbourne – Australia in 2014. He received his Master in Information Technology (Computer Network) from La Trobe University in 2009. He has published several international Journal and Conference papers. His research interests include business process modelling, business process reengineering, information system, security, business and IT alignment, software engineering, system analysis and design, sustainability and smart cities development.