

Crowdsourced Requirements Engineering Challenges and Solutions: A Software Industry Perspective

Huma Hayat Khan^{1,*}, Muhammad Noman Malik², Youseef Alotaibi³, Abdulmajeed Alsufyani⁴ and Saleh Alghamdi⁵

¹Department of Software Engineering, National University of Modern Languages, Islamabad, Pakistan

²Department of Computer Science, National University of Modern Languages, Islamabad, Pakistan

³Department of Computer Science, College of Computer and Information Systems, Umm Al-Qura University, Makkah, Saudi Arabia

⁴Department of Computer Science, College of Computers and Information Technology, Taif University, PO Box 11099, Taif, Saudi Arabia

⁵Department of Information Technology, College of Computers and Information Technology, Taif University, Taif, Saudi Arabia

*Corresponding Author: Huma Hayat Khan. Email: hnauman@numl.edu.pk

Received: 04 January 2021; Accepted: 01 March 2021

Abstract: Software crowdsourcing (SW CS) is an evolving software development paradigm, in which crowds of people are asked to solve various problems through an open call (with the encouragement of prizes for the top solutions). Because of its dynamic nature, SW CS has been progressively accepted and adopted in the software industry. However, issues pertinent to the understanding of requirements among crowds of people and requirements engineers are yet to be clarified and explained. If the requirements are not clear to the development team, it has a significant effect on the quality of the software product. This study aims to identify the potential challenges faced by requirements engineers when conducting the SW–CS based requirements engineering (RE) process. Moreover, solutions to overcome these challenges are also identified. Qualitative data analysis is performed on the interview data collected from software industry professionals. Consequently, 20 SW–CS based RE challenges and their subsequent proposed solutions are devised, which are further grouped under seven categories. This study is beneficial for academicians, researchers and practitioners by providing detailed SW–CS based RE challenges and subsequent solutions that could eventually guide them to understand and effectively implement RE in SW CS.

Keywords: Software crowdsourced; requirements engineering; software industry; software development; survey; challenges

1 Introduction

The crowdsourcing (CS) concept has arisen from new-fangled collaboration technologies such as social media and Web 2.0. The term CS was invented by Howe and Robison [1], who reported it as a fragment of notions such as mass collaboration, open collaboration and collective intelligence. It is a novel form of effort, in which a ‘crowd’ collaborates and accomplishes software development life cycle tasks (e.g. requirement



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

elicitation and analysis as well as design, coding and testing). This software development paradigm has emerged as an alternative to traditional software development organisations. Despite that the general effect has been ordinary so far, software crowdsourcing (SW CS) has the potential to and will enforce troublesome changes in how software development will be undertaken in the future [2–4].

SW CS is the assignation of a worldwide pool of online developers who can be appointed on-demand to contribute to several types of software development tasks [5–7]. The process is facilitated by platforms that link requesters and online developers. On this platform, the requester disseminates tasks to volunteer online developers, who solve the tasks based on their motivation for reward (e.g., money or respect). The SW CS platform has significant importance because it provides guidelines for managing and coordinating the processes and people in business and at technical levels [5,8]. Moreover, the platform permits requesters to discover talent outside their borders and earn the benefits of cost, quality, time and expertise [3,5,6]. In essence, this reveals that the software development team encounters an extensive user audience, which is referred to as the crowd of people. It is important to involve the crowd to satisfy crowd requirements, which demonstrates the importance of crowdsourcing [9]. Involving an enormous number of users in requirements engineering (RE) has always been challenging with customary RE methods [10,11]; specifically, this is true when RE would include an enormous number of users (a crowd) who are beyond an organisation's reach [10,12].

Traditional approaches to conducting the RE process typically involve an inadequate number of representatives in requirement-gathering sessions through interviews or focus groups [13]. Conversely, recent RE approaches, which are functional in market-driven RE, permit organisations to directly interact with main stakeholders who employ *ad hoc* feedback-gathering networks [14]. However, these recent RE approaches miss the prospect of uninterruptedly involving diverse groups of users who share their feedback through a variety of media [12,15,16]. Consequently, it is believed that the software development team cannot consider the varied circumstances of user subgroups when evolving the product [17,18]. So, valued resources for RE continue to be unexploited and software products might not fulfill users' needs [2,11].

CS-based RE is an umbrella term used for computerised or semi-computerised approaches to elicit and analyse information from a crowd to stem authenticated user requirements [19]. Usually, the crowd is an undefined set of people [5]; however, in the context of CS-based RE, a crowd is usually a large group of current or prospective software users who act together or with the software organisation's representatives (e.g., product owner or development team) [11]. These users are involved at the run time for requirement elicitation through users' feedback [20]. CS is considered a promising paradigm for eliciting requirements of a software system in a dynamic, possibly unknown context with a large pool of users [21]. CS-based RE is likely to enhance the quality, inclusiveness and even monetary viability of requirements elicitation. It allows software development teams to receive more updated knowledge about users' perceptions of the system's role to fulfil their requirements [9].

Although CS has noteworthy possibilities for RE, there remains scarce knowledge regarding how to set-up a CS-based project to fit the RE task [9]. A study conducted by Ågerfalk et al. [22] also specifies the importance of identifying various challenges that can be faced in SW CS. Further, Ågerfalk et al. [22] reported the importance of giving more attention to RE challenges to effectively implement the SW CS. Given the large effect of the challenges faced in the CS environment, it is difficult to form CS if the quality of elicited requirements is poor or unclear [11]. Therefore, this paper aims to investigate the potential challenges faced by the software development team in conducting SW–CS based RE. In addition, the solutions to overcome these challenges are investigated. The paper is outlined below. In Section 2, the related work is presented and in Section 3, the research method to achieve the research aim is reported. Section 4 discusses the research results, Section 5 provides an overall discussion of

the research and Section 6 reports the research limitations. Last, in Section 7, the research is concluded and the future work is described.

2 Related Work

It is essential to closely observe the conventional requirements gathering process to report the attributes of a software RE process conducted based on crowdsourcings. The RE process execution involves gathering, analysing, specifying and validating the requirements from stakeholders. In RE, multiple team members repeatedly converse with stakeholders in regard to requirements and continue to do so until a mutual agreement is reached on gathered requirements [23]. This conversation is conducted in formal meetings between the development team and stakeholders, in which secrecy remained until the software's first version was released [23]. In its conventional mode, RE relates to the system development that principally suits the requirements of its stakeholders (i.e., the financial owner of the system); however, because customers' (financial owners) eventual objective is that their system is used by end-users, their instant requirements should be reflected on during RE [24].

In CS-based RE, it is unlikely that requirements for developing a new system or version of an existing system are based on the crowd's perspective in an open call. A study conducted by Howe [1] first introduced the term CS as 'any acts a company or institution taking a function once performed by employees, and outsourcing it to an undefined (and generally large) network in the form of an open call' [23]. In this context, open call signifies that anyone who is interested might participate in the development. This type of process is made possible with the continuance of the internet. Stakeholders are the crowd in the CS environment and thus software development relies on them to gather their needs properly [9]. This indicates that it is time to rethink requirements elicitation to handle the difficulty and scale of the crowd and certify that the requirements are gathered efficiently and precisely [9]. The crowd could be the prospective users of the developed software to encounter their requirements; therefore, it is necessary to consider the crowd through the entirety of the software development process [9].

Several studies attempted to use the abilities of the crowd and end-users to resolve RE problems. It was found that these studies primarily focused on research areas such as requirements-driven social adaptation [20,25], feedback-based RE [26], stakeholders' discovery [27] and general requirements identification [28]. Zhang et al. [20,25] reported the difficulty of validating software in a dynamic context. According to the researchers, the unpredictability of software's role in fulfilling its needs in a dynamic context means that validating the software is a complex, difficult and time-consuming task. Ali et al. suggested the concept of social adaptation and social sensing for long-term validation of dissimilar alternatives to changing software. According to the researchers, it is an approach footed on obtaining and analysing users' awareness of the system's role in attaining their needs and its quality. The researchers have proposed to use this approach to generate adaptation decisions.

Similarly, research on feedback-based RE has been conducted. Martens et al. [26] discussed users' feedback. According to Pagano and Maalej, it is important to obtain users' software feedback to assist the development team in understanding the requirements of its future releases. The researchers suggested the effects of user feedback on RE teams by basing the significance of user feedback on the number of times a mobile application was downloaded. Other research [27] has reported CS from the perspective of stakeholders' discovery. This research has suggested that it is difficult to identify the stakeholders and their roles, skilfulness and needs for any complex and dynamic system. The study concluded that CS could assist in identifying the relevant stakeholders specified by the analyst. The study argued about the complexity behind identifying the relevant stakeholders and suggested a participatory approach to identify stakeholders. The work appraised stakeholders as a collaborative network. It is reported that by this, analysts who know only a few members would nominate additional members as analysts.

Further, previous literature has specified appropriate requirements identification as another issue. In various software paradigms, users are usually diverse and unpredictable. It becomes costly to rely on an elite set of users for the functionality and quality attributes of the software. CS could assist in tackling the crowd's capability to comprehend their requirements in a more desirable way. CrowdREquire [28] is a representative case of initiatives in which the notion of CS was supported for requirements elicitation. Existing studies have significantly reported the importance of RE in SW CS; however, they reported the need to give more attention to RE challenges to implement the SW CS effectively. Therefore, this research attempted to address this need by investigating the challenges faced by software development teams to conduct RE in SW CS. Moreover, the study also presents the suggested solutions to overcome these challenges.

3 Research Method

Qualitative research was conducted in this study because it is a type of scientific research that answers questions, gathers evidence, yields findings that were not found in advance and generates findings that are relevant outside the study's immediate limitations. Qualitative research is especially effective in gaining culturally detailed data about the values, opinions, behaviours and social contexts of specific populations [29]. This study uses the 'in-depth interview' technique to exploit the characteristics of qualitative research.

3.1 In-Depth Interviews

In-depth interviewing is a qualitative research technique that encompasses thorough individual interviews with a fewer number of respondents to discover perspectives of a particular idea or situation. In-depth interviews are valuable when comprehensive information is required about a person's opinion or behaviour or for discovering and discussing novel issues in depth [30]. The in-depth interviews were conducted according to the guide provided by Deterding et al. [30]. The method for conducting an in-depth interview involves the following process: plan, develop instruments, collect data, analyse data and disseminate findings [30]. Further thorough steps are described below.

3.1.1 Plan

The in-depth interview plan comprised the following primary tasks. First, stakeholders were identified. The survey invitation was delivered to various software development companies to identify the most relevant stakeholders. The Pakistan software engineering board database was used to select the companies. A total of 12 companies were contacted and six responded to the invitation. These companies exist in all major Pakistani cities; however, the head offices are situated in Islamabad, Lahore and Rawalpindi. The selected software development companies that participated in this study are displayed in [Tab. 1](#).

In total, 73 respondents from the selected companies were willing to participate in the survey. Following this, an appointment was scheduled with the willing respondents to conduct the interview. However, only 50 respondents were interviewed—23 respondents could not give their responses because of their work commitments. In this research, the participating stakeholders (respondents) were requirements engineers (also called business analysts) who worked in crowdsourcing platforms. Once the stakeholders were listed, the interview questions were established. For this research, stakeholders were asked about the challenges they face when conducting CS-based RE and the strategies used to overcome those challenges.

3.1.2 Develop Instrument

This stage involved developing the interview instrument. Initially, an interview protocol was developed. This protocol contained the guidelines that were followed for each interview to certify uniformity between interviews and thus increase the trustworthiness of the findings. Moreover, an interview guide was generated, which contained the interview questions. The instrument was validated for face validity and content validity

through two methods: Average congruency percentage (ACP) and content validity index (CVI). The instrument was validated through four experts who had educational, research and industry background in CS-based RE. All experts were specialised in crowdsourcing and had more than 10 years of industry and academic experience.

Table 1: List of companies

Company names	Requirement analysts	Software developers	Business analyst	Requirements engineers
Constant Variables	1	2	3	1
Average year of experience	7	6	8	4
Smart IS	1	3	3	2
Average year of experience	6	9	6	4
Adlabs	1	3	3	1
Average year of experience	4	7	5	5
One Bite	2	2	3	2
Average year of experience	4	8	4	6
Ninesol Technologies	1	3	3	2
Average year of experience	7	5	5	5
Fortlogics	2	2	3	1
Average year of experience	4	5	6	4

In ACP, experts computed the percentage of questions deemed to be relevant. Conversely, CVI calculated the content validity index for the individual item (I-CVI). Experts 1 and 3 found one out of 12 questions irrelevant, which resulted in a 91.66% relevancy rate at their level. However, Experts 2 and 4 rated all questions relevant, which resulted in a 100% relevancy rate at their level. The average value of the experts' congruency percentage was 95.5%, which was considered valid. For CVI, the four experts were asked to evaluate each question's content relevancy on a 4-point Likert scale, in which 1 = not relevant, 2 = somewhat relevant, 3 = relevant and 4 = very relevant. To decide the relevancy criteria for each question, the number of experts who gave three or four scores was counted as relevant and one or two scores were considered not relevant. It was found that the majority of questions (Q2, Q5, Q7–Q12) scored an I-CVI value of 1.00 (Q1, Q3–Q4 and Q6 scored an I-CVI value of 0.75). The resultant mean I-CVI value was 0.9, which was considered valid. Moreover, the proportion relevancy of questions was also calculated. It was found that all the experts rated the questions with high proportion relevancy, which resulted in a mean expert proportion of 0.87 (considered high). Based on the results, the face and content validity of the questions asked in the interview session were found to be significantly high, which ensured the quality of the instrument.

3.1.3 Train Data Collector

At this stage of the in-depth interviewing technique, interviewers were identified and trained. Two final-year research students of a bachelor's in software engineering programs were trained and involved in the data collection process. Irrespective of the data collectors' prior experience, comprehensive training was provided over two weeks. The training process comprised an introduction to the evaluation objectives, an analysis of

data collection techniques, a detailed analysis of the data collection items and instruments, preparation in the instrument usage, skill-building drills on interpersonal communication and conversation of ethical issues.

3.1.4 Collection of Data via Interview

Interviews were conducted with the 50 stakeholders (respondents). The interviews were conducted in groups with respondents from each participating company. First, interviewees' consent was taken and re-explained according to the interview's purpose, interviewees were notified as to why they were selected, and the interview session's expected duration was indicated. Once the interview was conducted, the interview minutes were summarised in the form of an interview report—a process that was completed for each interview within 2–3 h after the interview session.

3.1.5 Analyse Data

The gathered data were transcribed and analysed through grounded theory [31] data coding steps. The steps for data coding are detailed in Fig. 1.

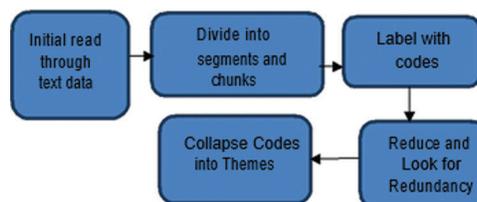


Figure 1: Data coding steps

The data were collected through interviews before being transcribed. Transcription is significant for qualitative research because it assists the researcher in analysing data easily and creating a narrative with the data. The transcription allowed the researchers to find data patterns and save the accuracy of the data. Initial reads were given to the data to achieve understanding regarding data. Once the researchers understood the data, they identified critical segments and assigned codes. Every response was allocated with a letter code, which was next entered into the software. Afterwards, the similar codes were grouped and redundancies were removed. Consequently, themes were generated and categorised according to their similarities. The researchers observed the created codes and identified various patterns, which resulted in themes. These themes are wider than codes. Researchers combined various codes into a unique theme. In this study, the codes represented the challenges and their solutions and the categories represented the various RE phases. Tab. 2 illustrates the detailed list of CS-based challenges and their solutions under the category of various RE phases.

4 Results and Discussion

In this section, the detailed result of the identified challenges and suggestions for SW CS are discussed. Moreover, the categories based on similarities and mapping of these challenges to RE phases are explained.

4.1 In-Depth Interviews

In classical software development, RE is critical; rather, it becomes more decisive in SW–CS based RE. To respond to the complexity of RE in SW CS, this research identified 20 challenges that requirements engineers face when conducting various phases of the RE process. Tab. 2 demonstrates the identified CS-based RE challenges and their suggested solutions for each RE phase. Tab. 2 comprises four columns: RE phase, challenges, challenge categories and suggested solution. In the ‘challenge’ column, each challenge was given an ID, which are referred to in subsequent sections.

Table 2: SW–CS based RE challenges and suggested solutions

RE phase	Challenges	Challenge category	Suggested solution
Requirement elicitation	Time-consuming to gather the requirements [c1]	Time	Time should be managed by proper planning of the requirements elicitation phase.
	Hard to obtain quality work [c2]	Quality work	Requirements engineers must define protocols to obtain quality work.
	Require more experts and resources [c3]	Experts required	More experts should be hired who are capable of handling requirements elicitation.
	Lack of confidentiality and communication [c4]	Privacy communication issues	Disclosure information should be enhanced to support privacy. Communication tools should be used for frequent communication.
Requirement analysis and design	Difficulty in the effective interpretation of user requirements [c5]	Requirements understanding because of a lack of crowdsourcing knowledge	Requirements engineers should consider the users' perspective to understand the requirements more effectively.
	Lack of conformity between the idea and solution [c6]	Conformity	The idea should have a context or purpose to conduct requirements analysis.
	Lack of face-to-face communication, which results in changing requirements [c7]	Lack of face-to-face communication	Use online tools to conduct online interviews and gather complete requirements at the start of the project, which would avoid changes in the future.
	Difficulty in user requirements fulfilment [c8]	Quality work	More than one RE technique should be used to have a detailed understanding of requirements for their complete fulfilment.
	Difficulty in designing requirements because people give requirements according to their culture and needs [c9]	Cultural issues	Filter the requirements according to users' needs to gain knowledge of users' cultures and moral values.
	Designed requirements conflict with user perspectives [c10]	Culture	Design requirements from the perspective of a user.
	Difficult to prepare, initialise, decompose and aggregate requirements [c11]	Requirements understanding because of lacking crowdsourcing knowledge	Platforms such as Crowd REquire should be used to conduct the RE process.
	Requirements engineers do not have the complete knowledge of the requirements in a crowdsourced environment, which leads to difficulty in requirements design [c12]	Requirement understanding because of lacking crowdsourcing knowledge	Platforms such as Crowd REquire should be used to conduct the RE process.

(Continued)

Table 2 (continued).			
RE phase	Challenges	Challenge category	Suggested solution
Requirement specifications	Difficult to manage and decide in finalising the requirements of numerous people [c13]	Requirement understanding because of a lack of CS knowledge	Enhance communication with clients and discuss their ideas with deeper attention. The communication mediums play a vital role in effectively discussing the problems and their solutions. The selection of the appropriate communication medium is vital to finalise the requirements among the stakeholders and discuss the shortcomings in the gathered requirements.
	Designing Software Requirement Specification (SRS) is complex because stakeholders are not on common ground [c14]	Complexity	Requirements engineers and customers must be on common ground.
	Requires more experts [c15]	Experts required	Experts who are fit for projects and can deliver in time should be hired.
	Lack of face-to-face communication between requirements engineers and clients [c16]	Confidence and confidentiality	Enhance communication with clients and discuss their ideas with deeper attention.
Requirement validation	Difficult to maintain a proper document presented according to the customer's needs [c20]	Quality work	Requirements engineers should check the quality of the drafts alongside each other by specifying them.
	Lack of quality assurance [c17]	Quality work	Targets should be achieved carefully and under serious observation.
	Difficult to obtain requirements for feedback from numerous people, which invalidates the requirements [c18]	Conformity	Requirements engineers should take precautions on earlier stages. A selected sample of people should perform validation.
	Time-consuming to manage or validate the gathered requirements from the stakeholder [c19]	Time	Requirements engineers should take precautions on the earlier stages so that there is no need to do so afterwards.

As described in [Tab. 2](#), the aforementioned challenges are faced by industry practitioners when conducting the RE process in a crowdsourcing platform. In total, 20 challenges were identified, which were further grouped under seven categories against each RE phase. In addition, the solutions of these challenges were described in detail. Further, the challenges are mapped with four RE activities: requirement elicitation, requirement analysis and design, requirement specification and requirement validation. For example, the activity 'requirement elicitation' involves four challenges that respondents most commonly face, and the respondents have suggested the most appropriate solutions overcome these challenges.

It was reported that eight challenges were most faced by requirements engineers when they performed requirement analysis and design:

- difficulty in the effective interpretation of user requirements [c5]
- lack of conformity between the idea and solution [c6]
- lack of face-to-face communication, which results in changing requirements [c7]
- difficulty in user requirements fulfilment [c8]
- difficulty in designing requirements because people give requirements based on their culture and needs [c9]
- designed requirements conflict with user perspectives [c10]
- difficult to prepare, initialise, decompose and aggregate requirements [c11]
- requirements engineers do not have the complete knowledge of the requirements in a CS environment, which leads to difficulty in requirements design [c12].

Subsequently, the respondents provided solutions to overcome these challenges. The respondents noted that, when they performed requirements specification activities, they predominantly encountered five challenges:

- difficult to manage and decide in finalising the requirements of numerous people [c13]
- designing SRS is complex because stakeholders are not on a common ground [c14]
- requires more experts [c15]
- lack of face-to-face communication between requirements engineers and clients [c16]
- difficult to maintain a proper document according to the customer's needs [c20].

Solutions to overcome these challenges are also described in detail. Last, requirement validation included three common challenges:

- lack of quality assurance [c17]
- difficult to obtain requirements for feedback from numerous people, which invalidates the requirements [c18]
- time-consuming to manage or validate the gathered requirements from the stakeholder [c19].

4.2 Categories of Software–Crowdsourcing Based Requirements Engineering Challenges

The research results (challenges) were analysed thoroughly and grouped into various categories based on their similarity. In particular, there were 20 challenges, in which each challenge pertinent to a specific aspect was grouped under a category. Fig. 2 depicts the categories of the identified RE challenges for SW CS.

As demonstrated in Fig. 2, the challenges are mapped in terms of their challenge ID to the seven categories:

- ‘Time-consuming to gather the requirements [c1]’, and ‘time-consuming to manage or validate the gathered requirements from the stakeholder [c19]’ are challenges related to ‘time’.
- ‘Hard to obtain quality work [c2]’, ‘difficulty in user requirements fulfilment [c8]’, ‘lack of quality assurance [c17]’ and ‘difficult to maintain a proper document presented according to the customer's needs [c20]’, are challenges related to ‘quality work’.
- ‘Difficulty in designing requirements because people give requirements according to their culture and needs [c9]’ and ‘designed requirements conflict with user perspectives [c10]’ are challenges related to ‘culture’.
- ‘Require more experts and resources [c3]’ and ‘requires more experts [c15]’, are categorised under ‘require more experts’ because both relate to the aspect of lacking experts’ comprehensiveness.

- ‘Lack of face-to-face communication, which results in changing requirements [c7]’, ‘designing SRS is complex as stakeholders are not on a common ground [c14]’, ‘lack of face-to-face communication between requirements engineers and clients [c16]’ and ‘lack of confidentiality and communication [c4]’ are challenges related to ‘communication and confidentiality’.
- ‘Lack of conformity between the idea and solution [c6]’ and ‘difficult to obtain requirements for feedback from numerous people, which invalidates the requirements [c18]’ are categorised under ‘conformity’ because both relate to the aspect of lacking conformity between requirements.
- ‘Difficulty in the effective interpretation of user requirements [c5]’, ‘difficult to prepare, initialise, decompose and aggregate requirements [c11]’, ‘requirements engineers do not have the complete knowledge of the requirements in a CS environment, which leads to difficulty in requirements design [c12]’ and ‘difficult to manage and decide in finalising the requirements of numerous people [c13]’ are categorised under ‘requirement understanding based on a lack of CS knowledge’ because the challenges occur from a lack of CS knowledge.

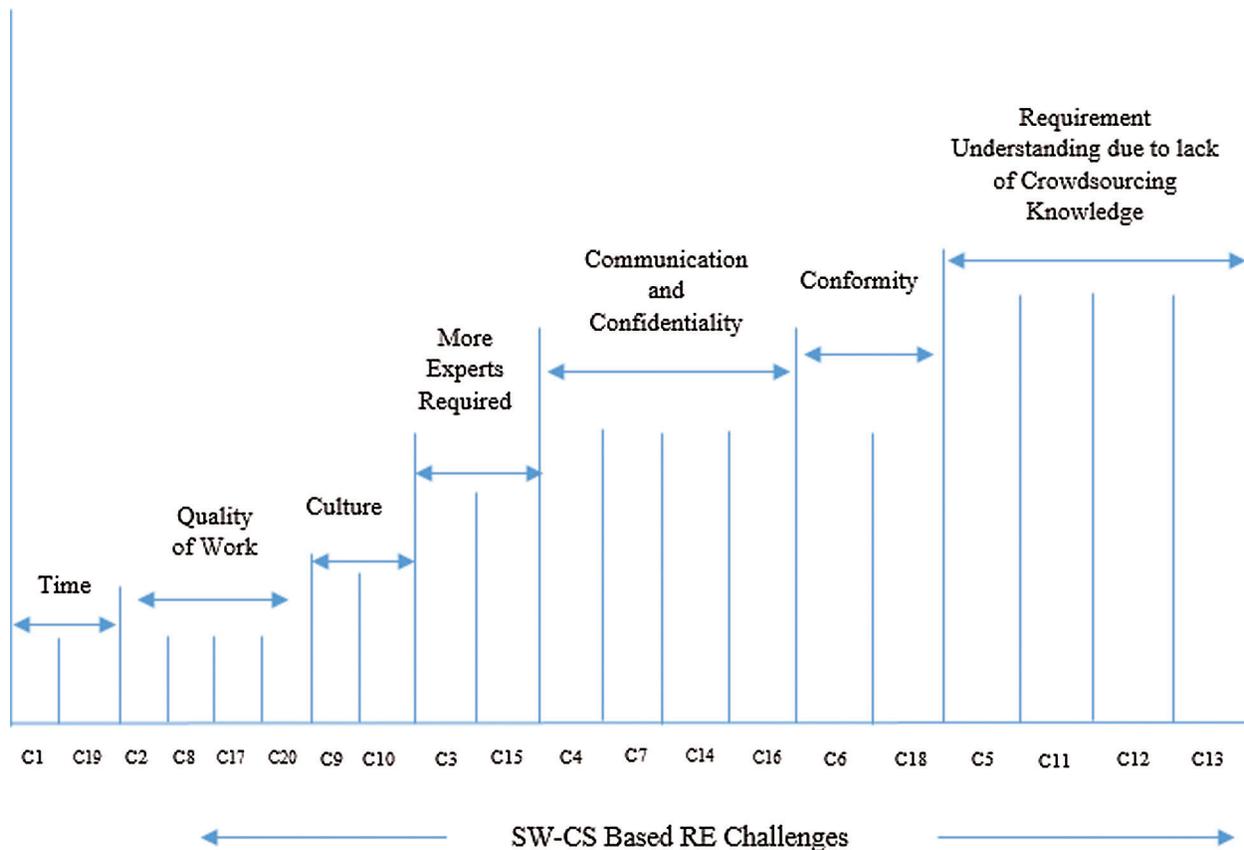


Figure 2: Categories of SW–CS-based RE challenges

4.3 Mapping the Software–Crowdsourcing Based Challenges with the Requirements Engineering Process

The research findings also generated mapping patterns between the RE process and challenge categories. Tab. 3 explains the identified SW–CS based challenges categories and their occurrence at RE activities.

As demonstrated in Tab. 3, identified challenges grouped under their respective categories occurred at various RE process activities. According to the respondents, challenges regarding ‘time’ were most common when conducting requirement elicitation and requirement validation. Challenges regarding ‘quality work’

appeared to be more profound throughout the RE process. Moreover, challenges regarding ‘more experts required’ commonly occurred when analysing, designing and specifying the requirements. Last, challenges regarding ‘confidence and confidentiality’ are usually faced when requirements engineers were involved in requirements elicitation and analysing, designing and specifying the requirements.

According to the respondents, the challenges regarding ‘conformity’ occurred while the RE team analyses, designs and validates the gathered requirements. Further, the challenges regarding ‘requirement understanding based on a lack of CS knowledge’ were faced by most requirements engineers when they performed requirement analysis, design and specification. To increase an in-depth understanding and visualisation of the findings, [Fig. 3](#) depicts a diagrammatic representation of the most commonly occurring RE challenges in CS at each RE activity.

Table 3: Mapping SW–CS based challenges categories with RE process

RE activities	Time	Quality work	Culture	More experts required	Confidence and confidentiality	Conformity	Requirement understanding based on a lack of CS knowledge
Requirements elicitation	X	X		X	X		
Requirements analysis and design		X	X		X	X	X
Requirements specification		X		X	X		X
Requirements validation	X	X				X	

5 Focus Group

The focus group was conducted to validate the research findings (SW–CS based RE challenges and solutions). Jay’s [32] study was followed to conduct a focus group. [Fig. 4](#) depicts the detailed process that was followed to conduct the focus group sessions.

The detailed focus group process consisted of six steps. First, it was necessary to define the focus group’s objective and validate the outcome of this research by comprising CS-based RE challenges and their solutions. The second step involved establishing a timeline. Planning the focus group session began six weeks before the session occurred. Eight senior requirements engineers were contacted, each with at least 10 years of experience, and six consented. Next, formal invitations were delivered. These six experts were serving in the industry as ‘principal software engineer’, ‘software project manager’, ‘senior software developer’ and ‘senior team leads’. It was ensured that all experts had been working in a CS environment. The interview questions were prepared thoroughly (see [Tab. 4](#)).

As demonstrated in [Tab. 4](#), the sample questions were asked in the main session of the focus group. The facilitator—a senior researcher—opened the session. First, the facilitator welcomed all participants, reviewed the purpose of the focus group and the aims or objectives of the meeting. The facilitator ensured that everything flowed smoothly in the meeting by establishing the ground rules (e.g. how the session will continue and how the participants can contribute).

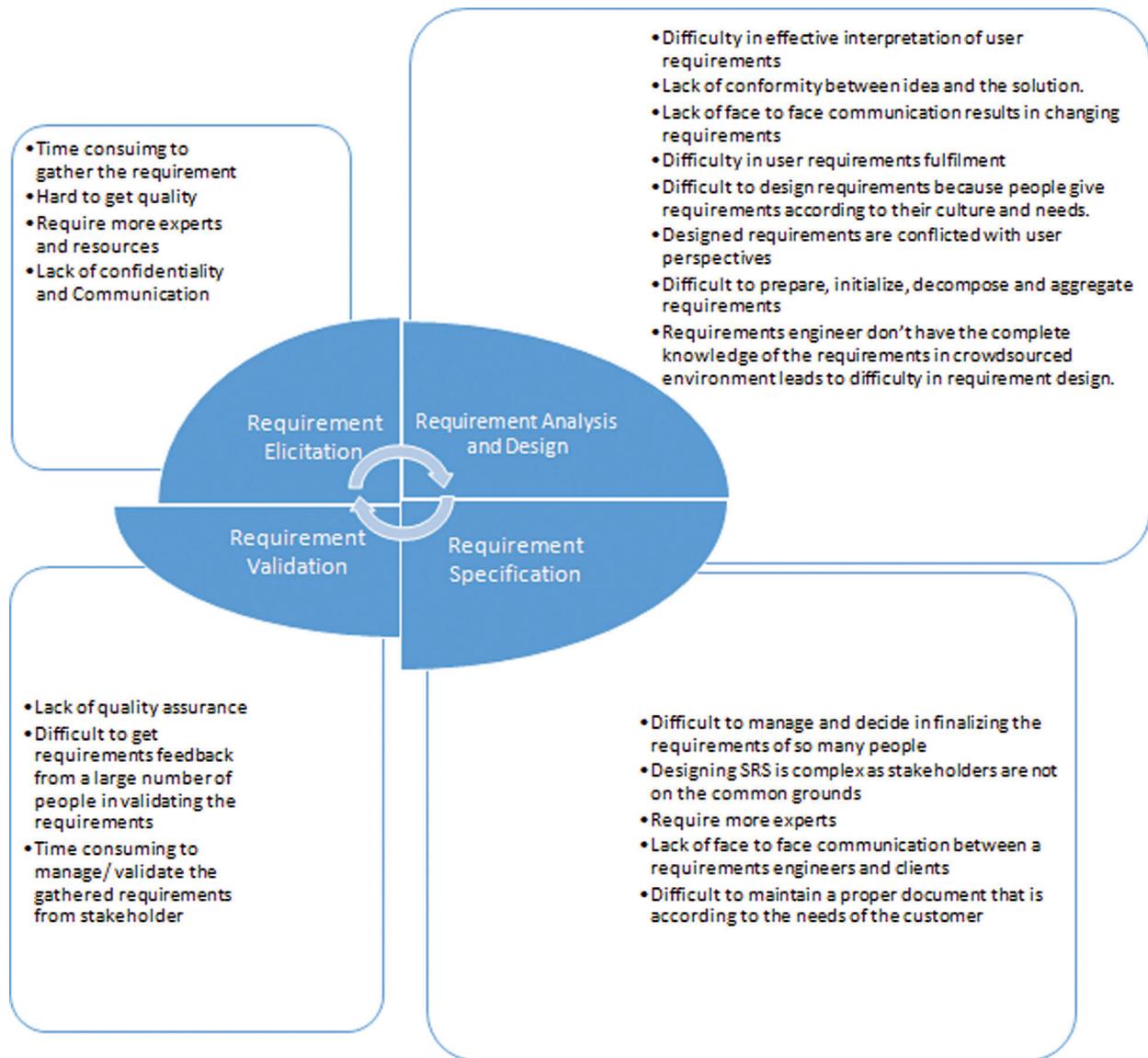


Figure 3: Commonly occurring challenges in the RE process

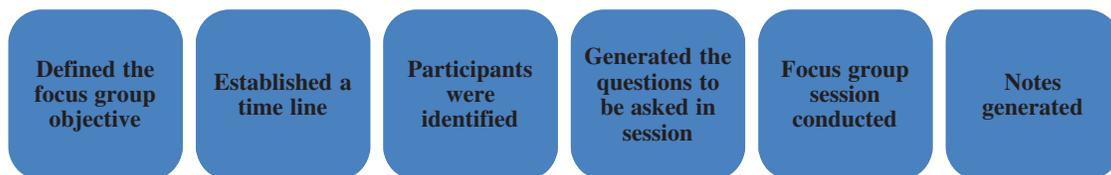


Figure 4: Focus group process

The session began with a general open-ended question—‘what do you think about SW–CS based RE?’. It was ensured that all opinions regarding various questions were conveyed and considered. The participants’ comments or feedback regarding the identified CS-based RE challenges and their solutions were recorded. The participants suggested some modifications in regard to the challenges and solutions; consequently, a

modified list of CS-based RE challenges and solutions was generated. [Tab. 2](#) displays the CS-based RE challenges and solutions. Other than the suggested modifications, the experts also commented on the study's comprehensiveness and strengths and weaknesses, as well as the comprehension of the recommended solutions and the barriers that restrict the adoption of the study.

Table 4: Sample of open-ended questions in the focus group session

Sr. No.	Questions
1	What do you think about CS-based RE?
2	Do you think that the proposed study is comprehensive in terms of CS-based RE challenges and their solutions? (How?)
3	What are the strengths and weaknesses of the research outcome (CS-based RE challenges and their solutions)?
4	Do you see any problem with the recommended solution comprehension?
5	In your opinion, what are potential barriers of adopting these suggested solutions?
6	What aspects of the suggested solutions can be improved and how?

5.1 The Comprehensiveness of the Software–Crowdsourcing based Challenges and Suggested Solutions

All the participants agreed on the study's comprehensiveness. It was noted that the study could act as a guide for academicians and practitioners working in the domain of CS-based RE. According to Expert 6, categorising challenges and mapping with RE phases would further the reader's understanding. Further, Expert 4 noted that the detailed solutions against the challenges enhance the study's comprehensiveness.

5.2 Strengths of the Study

According to the participants, the study findings were easy to understand and grasp. The experts reported the authenticity of the identified challenges by mentioning the comments of other session participants. There was a consensus regarding the practicality of the identified challenges and their solutions in a real environment. The experts suggested some modifications to the findings to enhance their understandability (see [Tab. 2](#)).

5.3 Barriers Against Adopting the Findings

Interestingly, the participants described various organisational and personal barriers in regard to adopting the study findings. The experts mentioned the problem of mindset in any organisation—organisations could be reluctant to consider these findings when working in CS-based RE. Further, in regard to personal barriers, there could be a lack of understanding or passion in understanding the challenges and their solutions for effective CS-based RE.

In addition, the organisation's employees may possess a reluctant attitude towards adopting novel things. All these barriers have significance in their context; however, it was concluded that these barriers could be easily managed if careful attention is applied.

6 Limitations

There are often multiple validity threats in survey-based studies. One of the difficulties with the survey is that it occasionally had a low response rate and the likelihood of individual biases. Previous literature has reported that thoughts and views obtained through a survey could be prejudiced and dissimilar from the

real-world population distribution. Another study limitation regards the instrument (questionnaire) that covered most aspects related to the potential challenges and solutions. The instrument was statistically analysed in terms of face and content validity tests and the results demonstrated the instrument's authenticity and comprehensiveness. However, it is possible that various significant questions that could have been further explored were overlooked.

The authors gathered qualitative data by conducting interviews with 50 respondents from the software industry who worked in a CS platform. In particular, the interviewees were all requirements engineers. Although their responses were quite comprehensive, the sample size is limited in terms of generalising the results. This limitation was attempted to be overcome through conducting a focus group, which would ideally validate and generalise the list of challenges and solutions. Last, the findings of this research is based on a human-understanding based data extraction strategy, which may have affected the outcome of this research. To address this limitation, experts were involved from both academia and the industry to ensure the contextual insights of the results.

7 Conclusion and Future Work

This research study has identified 20 SW-CS based RE challenges and their subsequent proposed solutions through an industrial survey. These challenges were further grouped into seven categories. The authors validated the findings by including eight experts in a focus group session. A total of 12 companies were contacted and six were willing to respond to the invitation. In total, 73 relevant respondents from the selected companies were willing to participate in the survey; however, only 50 respondents were interviewed because of attrition. SW CS exhibited a basic pattern swing in regard to how the software would be established in the future. Consequently, this has lifted numerous issues and RE has become one of the most pressing issues. In this regard, if RE is not properly handled in SW CS, the quality of the software work product will be greatly affected. In response, this study has involved a qualitative data analysis through conducting interviews with software industry professionals. Consequently, 20 SW-CS based RE challenges and their proposed solutions were devised. The factors were categorised under seven categories—time, quality work, culture, more experts required, confidence and confidentiality, conformity and requirement understanding based on lacking CS knowledge. Next, the research findings were validated through a focus group, in which experts evaluated the comprehensiveness of the findings, the strengths and weaknesses of the results and potential barriers against adopting the findings. It is believed that SW CS platforms can be assisted by this research in terms of obtaining a better understanding and gathering of software requirements. Researchers, academicians and practitioners can benefit from this research by utilising the research outcome. Moreover, future research could engage in the following activities:

- using empirical evidence to evaluate the proposed mapping between CS-based RE challenges and strategies to solve them
- using a wider audience to validate whether lists of challenges and solutions are exhaustive.

Acknowledgement: We deeply acknowledge Taif University for Supporting this study through Taif University Researchers Supporting Project number (TURSP-2020/115), Taif University, Taif, Saudi Arabia.

Funding Statement: 'This research is funded by Taif University, TURSP-2020/115'.

Conflicts of Interest: The author declares that they have no conflicts of interest to report regarding the present study.

References

- [1] K. Zaaout and K. Barker, "Structure of crowdsourcing community networks," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 1, pp. 144–155, 2018.
- [2] C. Wang, M. Daneva, M. van Sinderen and P. Liang, "A systematic mapping study on crowdsourced requirements engineering using user feedback," *Journal of Software: Evolution and Process*, vol. 31, no. 10, pp. 181, 2019.
- [3] T. D. LaToza and A. van der Hoek, "Crowdsourcing in software engineering: models, motivations, and challenges," *IEEE Software*, vol. 33, no. 1, pp. 74–80, 2016.
- [4] Z. Liao, X. Xu, P. Lan, L. Yang, Y. Zhang *et al.*, "A crowdsourcing recommendation that considers the influence of workers," *Computers, Materials & Continua*, vol. 66, no. 2, pp. 1379–1396, 2021.
- [5] 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, no. 2, pp. 57–84, 2017.
- [6] K. J. Stol, T. D. LaToza and C. Bird, "Crowdsourcing for software engineering," *IEEE Software*, vol. 34, no. 2, pp. 30–36, 2017.
- [7] S. Bibi, I. Zozas, A. Ampatzoglou, P. G. Sarigiannidis, P. G. Kalampokis *et al.*, "Crowdsourcing in software development: Empirical support for configuring contests," *IEEE Access*, vol. 8, pp. 58094–58117, 2020.
- [8] J. Troll, I. Blohm and J. M. Leimeister, "Why incorporating a platform-intermediary can increase crowdsourcees' engagement," *Business & Information System Engineering*, vol. 61, no. 1, pp. 1–18, 2019.
- [9] H. Guo, Ö. Kafalı, A. L. Jeukeng, L. Williams and M. P. Singh, "Çorba: Crowdsourcing to obtain requirements from regulations and breaches," *Empirical Software Engineering*, vol. 25, no. 1, pp. 532–561, 2020.
- [10] F. Cappa, F. Rosso and D. Hayes, "Monetary and social rewards for crowdsourcing," *Sustainability*, vol. 11, no. 10, pp. 2834, 2019.
- [11] S. A. Ebad and M. Ahmed, "Investigating the effect of software packaging on modular structure stability," *Computer Systems Science and Engineering*, vol. 34, no. 5, pp. 283–296, 2019.
- [12] Y. Alotaibi, "A new secured E-Government efficiency model for sustainable services provision," *Journal of Information Security and Cybercrimes Research*, vol. 3, no. 1, pp. 75–96, 2020.
- [13] J. Melegati, A. Goldman, F. Kon and X. Wang, "A model of requirements engineering in software startups," *Information and Software Technology*, vol. 109, no. 1, pp. 92–107, 2019.
- [14] Y. Alotaibi, "Automated business process modelling for analyzing sustainable system requirements engineering," in *2020 6th Int. Conf. on Information Management (ICIM)*, IEEE, London, UK, pp. 157–161, 2020.
- [15] Y. Alotaibi, "A new database intrusion detection approach based on hybrid meta-heuristics," *Computers, Materials & Continua*, vol. 66, no. 2, pp. 1879–1895, 2021.
- [16] X. Kong, X. Liu, B. Jedari, M. Li, M. Wan *et al.*, "Mobile crowdsourcing in smart cities: technologies, applications, and future challenges," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8095–8113, 2019.
- [17] I. Morales-Ramirez, F. M. Kifetew and A. Perini, "Speech-acts based analysis for requirements discovery from online discussions," *Information Systems*, vol. 86, no. 9, pp. 94–112, 2019.
- [18] T. Wang, T. Wu, A. H. Ashrafzadeh and J. He, "Crowdsourcing-based framework for teaching quality evaluation and feedback using linguistic 2-tuple," *Computers, Materials & Continua*, vol. 57, no. 1, pp. 81–96, 2018.
- [19] E. Guzman, R. Alkadhi and N. Seyff, "An exploratory study of Twitter messages about software applications," *Requirements Engineering*, vol. 22, no. 3, pp. 387–412, 2017.
- [20] D. Zhang, Y. Ma, X. S. Hu and D. Wang, "Towards privacy-aware task allocation in social sensing based edge computing systems," arXiv preprint arXiv:2006.03178, 2020.
- [21] M. Modaresnezhad, L. Iyer, P. Palvia and V. Taras, "Information Technology (IT) enabled crowdsourcing: a conceptual framework," *Information Processing & Management*, vol. 57, no. 2, pp. 102135, 2020.
- [22] P. J. Ågerfalk, B. Fitzgerald and K. J. Stol, *Software sourcing in the age of open: Leveraging the unknown workforce*. Springer International Publishing, Heidelberg, Germany, 2015.
- [23] Z. Hu, W. Wu, J. Luo, X. Wang and B. Li, "Quality assessment in competition-based software crowdsourcing," *Frontiers of Computer Science*, vol. 14, no. 6, pp. 74, 2020.

- [24] C. Burnay, "Are stakeholders the only source of information for requirements engineers? toward a taxonomy of elicitation information sources," *ACM Transactions on Management Information Systems (TMIS)*, vol. 7, no. 3, pp. 1–29, 2016.
- [25] L. Llerena, N. Rodriguez, J. W. Castro and S. T. Acuña, "Adapting usability techniques for application in open source Software: A multiple case study," *Information and Software Technology*, vol. 107, pp. 48–64, 2019.
- [26] D. Martens and W. Maalej, "Towards understanding and detecting fake reviews in app stores," *Empirical Software Engineering*, vol. 24, no. 6, pp. 3316–3355, 2019.
- [27] R. R. Schreiber and M. P. Zylka, "Social network analysis in software development projects: A systematic literature review," *International Journal of Software Engineering and Knowledge Engineering*, vol. 30, no. 03, pp. 321–362, 2020.
- [28] 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, no. 10, pp. 108–123, 2018.
- [29] C. Wagner, B. Kawulich and M. Garner, "A mixed research synthesis of literature on teaching qualitative research methods," *SAGE Open*, vol. 9, no. 3, pp. 215824401986148, 2019.
- [30] N. M. Deterding and M. C. Waters, "Flexible coding of in-depth interviews: A twenty-first-century approach," *Sociological Methods & Research*, pp. 1–32, 2018.
- [31] J. Corbin and A. Strauss, *Basics of qualitative research: Techniques and procedures for developing grounded theory*. SAGE Publications, Chicago, USA, 2014.
- [32] J. Klagge, "Guidelines for conducting focus groups," 2018 [Accessed 2021 January 24]. Available from [HTTPS://www.researchgate.net/publication/327607001](https://www.researchgate.net/publication/327607001).