

Research Efforts and Challenges in Crowd-based Requirements Engineering: A Review

Rosmiza Wahida Abdullah¹, Sabrina Ahmad^{2*}, Siti Azirah Asmai³

Fakulti Teknologi Maklumat Dan Komunikasi
Universiti Teknikal Malaysia Melaka
Malaysia

Seok-Won Lee⁴

Dept. of Software and Computer Engineering
Dept. of Artificial Intelligence
Ajou University, Suwon, South Korea

Zarina Mat Zain⁵

Digital Group
PETRONAS Digital Sdn. Bhd
Kuala Lumpur, Malaysia

Abstract—Eliciting software system development requirements is a challenging task as the information is from various resources. The most constructive resource is the stakeholders of the system to be developed. It is critical yet time-consuming to capture essential requirements to realize a reliable and workable software system. The crowd-based Requirements Engineering (crowd-based RE) approach adapts the crowdsourcing technique to access an extensive range of stakeholders and save time, especially for the generic type system with no clear stakeholder. This paper presents current research efforts and challenges in crowd-based RE. A systematic literature review method is adopted to explore literature based on two specific research questions. The first question aimed at identifying research efforts on crowd-based RE, and the second question focused on the main challenges discovered in pursuing crowd-based RE. The findings from the literature review show that many efforts have been made to explore and further improve crowd-based RE. This paper provides a foundation to pursue research in improving crowdsourcing techniques for the benefit of requirements engineering.

Keywords—Crowd-based requirement engineering; requirements engineering; requirements elicitation; software engineering; crowdsourcing; review

I. INTRODUCTION

Requirement Engineering (RE) is the first and the most crucial phase in a software development project, and the process must be performed to ensure quality software requirements. A study [7] stated that a poorly engineered requirements process contributes immensely to the failure of software projects. It is also said that projects that undermine RE suffer or are likely to suffer from failures, challenges, and other risks [36].

Requirements collected must be correct, complete, and concise to ensure the success of the developed software system. To do that, requirement engineers need to specify the stakeholders and ensure they participate in providing the requirements [3]. The process is challenging as it needs to gather and translate the imprecise, incomplete needs and wishes of the stakeholders into complete, precise, and formal

specifications. In the case of requirements from the crowd being welcome and deemed helpful in ensuring preferred system features are incorporated, the crowdsourcing technique is beneficial. The term crowdsourcing is introduced to portray the concept of outsourcing that describes a distributed problem-solving approach online with a large number of people [1]. Due to the advancement in Internet technology, crowdsourcing is now an emerging technique that has been actively studied and adapted in various domains such as software engineering, social innovation, and education. While the crowdsourcing technique is gaining popularity in multiple domains [2], requirements engineering should also benefit since crowdsourcing makes it possible to reach out to many stakeholders to voice out their needs and expectations towards a particular software system.

However, there is always a catch to benefit from such an emerging technique. While much information is good, issues like overloading, coverage, unknown source, and unreliable information need to be taken care of while eliciting requirements from the crowd. How are we going to ensure that the information we receive is enough? Is it from reliable resources? Is the information meaningful? Is it reliable? We have to deal with these challenges while eliciting requirements through crowdsourcing techniques since the Internet has no boundary.

Therefore, the ultimate aim of this paper is to provide an insight for further exploration and contribution towards strengthening crowd-based requirements engineering. The expected contributions of this research are:

- 1) To discover research efforts on crowdsourcing that have been done to empower requirements engineering in the years range from 2008 until 2021. We intend to discover the RE activities supported by crowdsourcing.
- 2) To present the chronology of research efforts to recognize issues and improvement in crowd-based requirements engineering thus far. The findings will be helpful to identify the research gaps that form a basis for future crowd-based RE research.

*Corresponding Author
(Universiti Teknikal Malaysia Melaka sponsors this paper through a research grant numbered PJP/2020/FTMK/PP/S01774)

Following the introduction, Section II explains the background of the study. This is followed by Section III, which elaborates on the systematic literature review method. Section IV presents the review results, and Section V elaborates on the discussion. Finally, Section VI concludes the paper.

II. BACKGROUND

Traditional RE adopts conventional techniques such as interviews, surveys, document analysis, workshops, and brainstorming. To elicit requirements through these traditional techniques are challenging and costly in term of time and effort. The chances to miss out on essential requirements from the key stakeholders are also very high due to resource constraints to implement adequate RE. Therefore, many types of research are conducted to enhance the user involvement in the RE process within limited resources [6].

In line with the available Internet technology and how information is exchanged nowadays, it is only reasonable that the RE techniques have evolved. Besides, people are now very much exposed to doing things online, from communication to paying bills and even controlling smart facilities from a distance. The rapid rise in Internet, mobile and social media applications makes it even more possible to provide channels to link a large pool of highly diversified and physically distributed stakeholders, especially potential users, for the system to be developed [5].

Crowdsourcing is an evolving paradigm that provides help to gather enormous and functional software requirements. Crowdsourcing makes it possible to reach out to many stakeholders to offer or voice their needs and expectations towards a particular software system. By adopting crowdsourcing, we reduce the risk of missing essential requirements from specific key stakeholders. In [1], J. Howe introduced the term crowdsourcing, adapted from the concept of outsourcing that describes a distributed problem-solving approach online with the involvement of a large number of people. In [34], M. Hosseini et al. mentioned four critical features of crowdsourcing: the crowd; people participating in the crowdsourcing activity, the crowdsourcer; the party that owns the task, the crowdsourcing task, and the crowdsourcing platform; the setting where the mission is accomplished. Crowdsourcing is gathering works, information, and opinions from the public through the Internet, social media, and smartphone apps [8]. According to U.S. Ghanyni et al. in [3], crowdsourcing offers a wide range of expertise and talents, making it the best way to collect requirements and improve user involvement. In 2015, the term Crowd-Based Requirements Engineering, also known as CrowdRE, was coined [10]. After that, in [9], E.C. Groen et al. defined CrowdRE as an umbrella term for automated or semi-automated RE approach for gathering and analyzing information from a crowd to derive validated user requirements.

Due to the numerous benefits of crowdsourcing, crowd-based RE is becoming popular and a meaningful way to be applied in the RE process, especially in requirement elicitation activity. This is because every stakeholder will get the opportunity to propose their expectations of the software [11].

Hence, the gathered requirements will be complete in representing sufficient stakeholders' perspectives and perceptions compared to limited input from selected stakeholders.

In line with that, J.A. Khan et al. in [5] supported the fact that there is a growing interest in crowd-based RE. Therefore, further research to improve the crowd-based RE is relevant for better service to the software engineering community.

III. METHODOLOGY

This section describes our literature survey process based on a systematic literature review method [12], which drives research questions through searching, filtering, and analysis processes. The literature exploration is presented through two research questions.

A. Research Questions

Due to the growing interest in crowd-based RE, this paper presents current research efforts and challenges to explore further opportunities to improve RE through crowdsourcing. The research questions addressed by this study are as follow:

RQ 1 What researches have been done in crowdsourcing for RE?

To answer RQ1, we conduct a literature review aimed at identifying research efforts on crowd-based RE.

RQ 2 What are the challenges and limitations of current research in crowd-based RE?

To answer RQ2, we look at the issues encountered in the recent crowd-based RE researches.

B. Search Process

This sub-section explains the searching strategies of this literature survey. The search is done manually through popular and familiar digital libraries and databases as listed below:

- 1) IEEE Xplore (ieeexplore.ieee.org).
- 2) ScienceDirect (sciencedirect.com).
- 3) Springer (Springerlink.com).
- 4) Google Scholar (scholar.google.com).
- 5) ACM Digital Library (dl.acm.org).

The searching included leading conferences, workshops, and journals that meet the search criteria. The search strings are based on the research questions and relevant keywords related to search areas such as requirements engineering, crowdsourcing, crowd-based and crowd-centric. We are aware that many articles about this topic are also posted on blogs, magazines, and newspapers, but we only focus on academic publications for this literature review. Besides, only papers written in English are covered.

C. Inclusion and Exclusion Criteria

Both research questions were answered by searching relevant research papers through meaningful keywords. The keywords from the primary studies were used to find more articles related to the research. Also, synonyms and alternative words were used to optimize the search of related works. The

general keywords used to search the associated articles were “crowd* AND requirement*.” We used the combination crowd* AND requirement* search term to ensure we managed to obtain as many relevant results as possible. The search gave us various reliable journals and conference proceedings covering issues in crowdsourcing for requirements engineering. Fig. 1 shows the number of research articles from 2008 until 2021. It shows the ascending pattern in crowd* AND requirement* search terms. Hence, we can conclude that many researchers are interested in this area, and crowdsource in RE is gaining popularity year by year.

Upon completing the searching process, we filtered the findings to related works only. We have included 20 primary studies that proposed approaches to automate the RE activities through crowdsourcing techniques. Fig. 2 shows the distribution over the years the studies have been published. Referring to our search, no effort has been proposed in 2009, 2013, and 2016, but the number of proposed efforts spiked in 2019. In 2020, however, only one proposed effort was discovered. Fig. 2 shows that researchers never stop exploring this area and proposed solutions that make use of the advantages in crowdsourcing to overcome or at least minimize problems in RE. Therefore, we believe that it is worth the effort to explore this area for the benefit of RE.

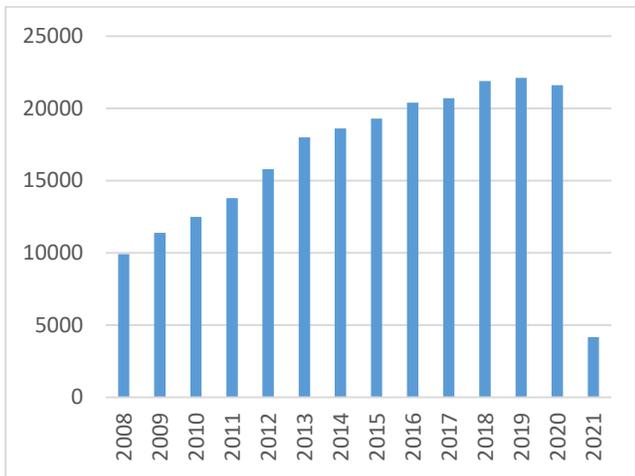


Fig. 1. Number of Research Articles.

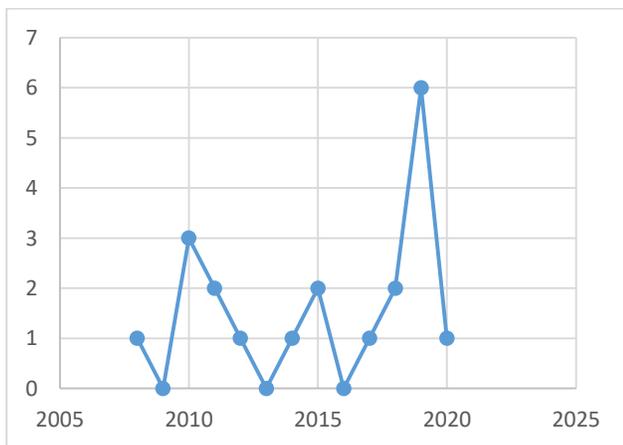


Fig. 2. Efforts on Approaches to Automate the RE Activities.

IV. REVIEW RESULTS

A. Research Question 1: What Researches have been done in Crowdsourcing for RE?

As we know, the ultimate idea of the crowd-based RE approach is to obtain input or feedback from the crowd who uses the software [35]. The crowdsourcing technique allows access to diverse stakeholders and can gain broader and up-to-date information about users’ expectations toward the system to be developed [33].

To answer this research question, we conduct a literature review to identify the RE activities supported by crowdsourcing. We reviewed 20 primary researches that related to crowd-based RE efforts. Table I provides an overview of the efforts of crowd-based RE approaches, and the table also covers the RE activities supported by the efforts.

TABLE I. CROWD-BASED RE APPROACHES/TOOLS/PLATFORM

| Approach/ Tool/ Platform | Description | Supported RE Activity | Source |
|--------------------------|--|--|--------|
| WikiWinWin | Wiki-based system for collaborative requirements negotiation | <ul style="list-style-type: none"> Elicitation Analysis Specification | [13] |
| StakeNet | Using social networks to analyze the stakeholders of large-scale software projects | <ul style="list-style-type: none"> Analysis | [14] |
| StakeSource | A web-based tool that automates stakeholder analysis | <ul style="list-style-type: none"> Analysis | [15] |
| StakeSource 2.0 | A web-based tool that utilizes social networks and collaborative filtering to identify and prioritize requirements | <ul style="list-style-type: none"> Analysis | [16] |
| iRequire | An application that gathers requirements to develop new mobile apps | <ul style="list-style-type: none"> Elicitation Analysis | [17] |
| StakeRare | Use social network and collaborative filtering for large-scale requirements elicitation | <ul style="list-style-type: none"> Analysis | [18] |
| CrowdREquire | A web-based platform that supports RE using crowdsourcing concept | <ul style="list-style-type: none"> Elicitation Validation | [4] |
| AppEcho | A user feedback approach for mobile platforms and applications | <ul style="list-style-type: none"> Elicitation | [19] |
| Refine | A platform that uses gamification for participatory requirements engineering | <ul style="list-style-type: none"> Elicitation Analysis | [20] |
| FAME | a framework for the combined and simultaneous collection of feedback and monitoring data in web and mobile contexts to support continuous requirements elicitation | <ul style="list-style-type: none"> Elicitation | [21] |

| | | | |
|--|--|-----------------------------|------|
| CRUISE | ➤ A platform for crowdsourcing Requirements Elicitation and evolution | • Elicitation | [22] |
| Effective classification methodology | An approach to classifying user requests into corresponding requirement types | • Elicitation • Analysis | [11] |
| GARUSO | ➤ a gamification approach for involving stakeholders outside organizational reach in requirements engineering. | • Elicitation • Analysis | [23] |
| CREeLS | A framework of the necessary elements of crowdsourcing suggesting specific tools for each component, and a phased approach to implementing the framework in the requirement elicitation activity for eLearning System | • Elicitation • Analysis | [24] |
| SUPERSEDE | provides advances on end-user feedback, contextual data analysis, and decision-making support in software evolution and adaptation | • Elicitation • Analysis | [25] |
| Classification model | The proposed classification model collects feedbacks from the crowd, and the feedback will be analyzed by applying data mining techniques. Finally, the proposed model will classify the feedbacks into functional and non-functional requirements. | • Elicitation • Analysis | [28] |
| Continuous Requirements-Elicitation Methodology | The proposed methodology captures and analyzes user feedback and comments on social networks such as Twitter for software systems in use and then extracts potential requirements. | • Elicitation • Analysis | [29] |
| Automated Feature Identification | The proposed solution is a tool-supported approach that analyzes apps and associated feedback on the feature level. This information is used to inspire the development of new apps and, in particular, to suggest features for developing similar new apps. | • Elicitation • Analysis | [31] |
| Crowdsourced RE Platform for User Story (US) Authoring | investigate how a crowdsourced RE platform can enable the crowd to provide requirements by using one of the RE artifacts' type, the US. | • Elicitation | [32] |

| | | | |
|------------|--|---------------|------|
| KMar Crowd | a CrowdRE platform called the KMar Crowd is applied in governmental organizations to discover the needs and wishes of user groups for a particular IT product. | • Elicitation | [30] |
|------------|--|---------------|------|

WikiWinWin was proposed by [13], which adapts the Wiki-based system that allows anybody to provide input to the platform. There are two types of primary users which are Shapers and Personal Knowledge Contributors (PKC). Shapers are the skilled stakeholders who contribute ideas, motivate PKC to express ideas, moderate the negotiation process, integrating, filtering, organizing, and rewriting contributions of others. While PKCs are the participants who contribute ideas and negotiate win conditions. The participants have to be invited to join in. At the end of the process, a software requirement description is produced.

StakeNet, StakeSource, and StakeRare concerned with stakeholder analysis. These tools are to carefully filter the stakeholder that participates in the project to contribute input for the requirements. Stakeholders are the source of the requirements, and we do not want to miss out on any crucial stakeholders to ensure the project's success [15].

iRequire and AppEcho enable mobile phone users to contribute feedback. These tools concern the end-user's involvement in obtaining the input and the context of the gathered information. For these tools, anybody who uses a mobile phone may participate in contributing the feedbacks.

Feedback Acquisition and Monitoring Enabler (FAME) also uses feedback and monitors the information to elicit new requirements. This tool is more to obtaining requirements for software evolution. FAME was developed as part of the SUPERSEDE EU project.

REFine, CrowdREquire, and CRUISE are concerned with stakeholder analysis and obtaining input from the invited stakeholder. Stakeholders can only participate through an invitation from the project owner. These tools incentivize the participant to motivate them to keep contributing to the project. REFine, and CRUISE applies game element while CrowdREquire offers a financial reward.

GARUSO approach uses a strategy for identifying stakeholders outside the organizational reach and a social media platform that applies gamification for motivating these stakeholders to participate in RE activities.

The SUPERSEDE (Supporting Evolution and Adaptation of Personalized Software by Exploiting Contextual Data and End-User Feedback; supersede.eu) project is developing multimodal-feedback functionalities that will let a crowd of users provide unobtrusive in situ feedback on software products. A runtime approach establishes comprehensive techniques to monitor software products and obtain environmental and context data through sensors. The received feedback and data will be analyzed to identify relevant information to support decision-making during software evolution. Informed decisions based on the feedback and monitoring data will lead to products that better meet user needs and improve the user experience.

The CREeLS, an effective classification methodology, and classification model are concerned with classifying the requirements. CREeLS adapt the approach proposed in the effective classification methodology, especially for the eLearning system. The classification model collects feedbacks from the crowd and then classifies the feedbacks into functional and non-functional requirements. CREeLS, effective classification methodology, and classification model apply text mining tools to analyze unstructured text because they use feedback as the input, which is usually natural language.

Continuous Requirement Elicitation Methodology and Automated Feature Identification also capture and analyze feedback from the crowd. Continuous Requirement Elicitation Methodology captures and analyzes user feedback and comments on social networks such as Twitter for a software system currently in use and then extracts the potential requirements. Automated Feature Identification examines feedbacks of mobile apps from the crowd to identify features for developing similar new apps.

Crowdsourced RE Platform for User Story (US) Authoring applies one of the RE artifacts: User Story. This research investigates how a crowdsourced RE platform can enable the crowd to provide requirements through four simple self-explanatory steps: Role, Goal, Benefit and Verification, and Category Selection.

KMar Crowd is a crowd-based RE platform applied in governmental organizations to identify the users' needs and wishes for a particular IT product. When this article is written, the researcher of KMar Crowd does not reveal how the platform works and what type of information is gathered from the crowd.

All the research efforts mentioned in this section applied diverse techniques and approaches to improve crowd-based RE in a specific area. More research should explore ways to utilize crowdsourcing and further improve RE.

B. Research Question 2: What are the Challenges and Limitations of Current Research in Crowd-based RE?

To answer this research question, we look into issues discovered in the current researches in crowd-based RE as listed in Table I. Through crowd-based RE, we can access a large pool of stakeholders to achieve the breadth of the requirements. However, some challenges need to be taken care of to guarantee the success of achieving the breadth of the requirements. As stated by D. Johnson et al. in [26], crowd-based RE has been argued to comprise four main activities: motivating crowd members, eliciting feedback, analyzing feedback, and monitoring context and usage data. These are essential elements to ensure that the information collected covers enough perspectives and to ensure if the information is reliable. In general, we found two main challenges in the existing research: stakeholders' coverage and information reliability.

The following sub-section is presented narratively to show efforts evolution to overcome the challenges:

1) *Stakeholders coverage*: Do we cover enough perspectives? The challenge here is whether or not the

information obtained represents enough perception and perspectives to develop a quality system to fulfill the system's purposes. Discussed below are research efforts to improve stakeholders' involvement to improve the coverage.

WikiWinWin, proposed by [13], provides a platform for the stakeholder to vote and decide on the software requirements. As the stakeholder participation only through invitation, the issue of missing key stakeholders is still there. Moreover, the stakeholder who participates in a specific project must understand well about the project they are participating in to make sure they provide ideas according to the context of the project. Other than that, the stakeholder needs to vote for the ideas to make it a requirement. If the idea is not getting many votes, it will not be considered a requirement. Thus, the stakeholders involved in the project must understand and be well aware of the expectation for the software to be developed. It is indeed crucial to establish the right system and, at the same time to fulfill the end-users need. Therefore, in WikiWinWin, the challenge is to select the right and sufficient stakeholders to participate. Besides that, it is also a challenge to keep the stakeholders motivated to provide ideas and input to the project.

Many software projects fail because they overlook stakeholders or involve the wrong representatives of significant stakeholders' groups [14]. Knowing the importance of obtaining correct stakeholders in the software development project, S. L. Lim et al. in [14] proposed a tool called StakeNet for stakeholders' analysis. StakeNet requires experts to identify stakeholders, and then, the experts have to ask them to recommend other stakeholders individually. Consequently, a social network of stakeholders based on their recommendations will be built. The prioritization of the stakeholders is decided by using various social network measures. However, this tool will be very costly for a large project which involves many stakeholders since it requires the experts to approach stakeholders individually to ask for recommendations.

Aware of this issue, StackSource was introduced [15]. StackSource is a web-based tool that automates stakeholder analysis. StackSource identifies stakeholders by asking them to recommend other stakeholders, builds a social network of stakeholders from their recommendations, and prioritizes them using social network measures. Soon after that, S. L. Lim et al. in [16] proposed an enhanced version of the StackSource tool StackSource2.0. Besides stakeholders' analysis, this improved tool is introducing another feature to do requirement elicitation and prioritization. In the requirements elicitation and prioritization feature, the tool can identify requirements by asking stakeholders to suggest and rate the requirements, recommend other requirements of interest using collaborative filtering, and prioritize the requirements using their ratings weighted by their priority in the social network.

Later in 2012, S. L. Lim and A. Finkelstein [18] proposed a method known as StakeRare that uses social networks and collaborative filtering to identify and prioritize requirements in large software projects. StakeRare identifies stakeholders and asks them to recommend other stakeholders and stakeholder roles, builds a social network with stakeholders as nodes and their recommendations as links, and prioritizes stakeholders

using various social network measures to determine their project influence. It asks the stakeholders to rate an initial list of requirements, recommends other relevant requirements using collaborative filtering, and prioritizes their requirements using ratings weighted by their project influence. Recently, an approach named GARUSO has been proposed [23] to identify stakeholders outside the organizational reach. It is a social media platform that applies gamification to motivate related stakeholders to participate in RE activities. Compared to StakeNet, StakeSource, Stake Source2.0, and StakeRare, GARUSO is claimed to reach potential stakeholders from multiple online channels such as e-mail services and SNSs to identify stakeholders of a software system who are beyond the reach of an organization.

CrowdREquire, REfine, and CRUISE adapt stakeholder analysis and requirement elicitation. Stakeholders can only participate through an invitation from the project owner. These tools provide incentives to motivate the participants to keep contributing to the project. REfine, and CRUISE adapts gamification while CrowdREquire offers a financial reward. Giving incentives and rewards to encourage the participation of the stakeholders may cause malicious and dishonest input. This is because the stakeholders might vote or offer information to gain reward, leading to incorrect requirements.

2) *Reliable information*: While much information is achievable through crowdsourcing, is the information useful? It is common knowledge that stakeholders especially end users are among reliable information sources from whom requirements are elicited. Traditionally, interviews, workshops, brainstorming, and survey will be conducted among the end-users to obtain requirements. It is clearly stated that software system users are an essential group of stakeholders, as reported by M. Bano and D. Zowghi [27]. End users' involvement in software development life cycle (SDLC) has been suggested to improve requirements' quality, accuracy, and completeness to ensure users' satisfaction. Presents below are research efforts that capture end-users input through crowdsourcing and introduce ways to ensure that the input is reliable.

Earlier in 2010, N. Seyff et al. in [17] stated that end-users involvement is particularly relevant for early software engineering activities such as requirements elicitation. In that particular study, iRequire is introduced to capture end-user requirements for mobile. In 2014, an app named AppEcho was introduced [19]. This app is a feedback approach that enables users to give feedback through the android platform. The method allows smartphone users to actively participate in continuous evolution and improvement by providing individual feedback to developers.

Furthermore, FAME was also introduced [21] to collect users' feedback on the software product. It is a stand-alone feedback app for mobile devices. FAME was developed as part of the SUPERSEDE EU project. SUPERSEDE project is a runtime approach to collect and analyze user feedback. It is also managed to identify relevant information to decide the essential requirements for the next release of a product. In 2019 Continuous Requirement Elicitation Methodology [29] and Automated Feature Identification [31] were proposed. Both of

these researches use feedbacks as the primary source for the information. A study conducted by A. Alwadin and M. Asharagi in [29] collected feedback and comments from the crowd via Twitter for an in-use software system. They applied data retrieval and natural language processing (NLP) techniques to extract potential requirements. Automated Feature Identification was proposed by T. Iqbal et al. in [31] to identify features for developing new mobile apps. This research applied the app store mining technique, exploring crowd-generated data such as feedback on the existing apps to identify critical elements for creating new apps. Machine Learning is used to analyze the input.

Another proposed solution introduced by C. Li et al. in [11] is a framework that allows information related to the software to be developed gathered from various sources, including users' feedback on SNS, previous project documentation, and experts. AI techniques are applied to process the collected information, and finally, the requirements descriptions are produced. Later, N. M. Rizk et al. in [24] adapted the methodology introduced by [11] to proposed CREeLS. CREeLS is offered specifically for the eLearning System. A classification model was proposed in another research conducted by S. Taj et al. in [28]. This model enables the crowd to actively participate in providing feedback which later, the feedbacks will be classified into functional and non-functional requirements.

In 2019, a study [32] proposed applying one RE artifact, User Story (US), in a crowd-based RE platform. It is reported that USs are estimated to be used by over half of the practitioners in the software industry to capture requirements. In this research, the participants are from the crowd. The participants have to provide information through four self-explanatory steps: Role, Goal, Benefit and Verification, and Category Selection. Finally, USs are formulated by the data the participants provided in these four steps. The requirement engineering will extract the potential requirements from the formulated USs.

V. DISCUSSION

In a study, Altug [35] stated that requirement engineering is a crucial stage in the software development life cycle process. During this stage, the requirement engineer must determine the optimum depth and breadth to obtain quality requirements. Crowd-based RE is an emerging approach that can help in securing quality software requirements. As depicted in Table I, we reviewed 20 primary researches that relate to crowd-based RE efforts.

In RQ1, we found that the researchers used no dominant techniques. However, we found that all 20 research efforts applied crowd-based RE in analysis, elicitation, or both. Analysis and elicitation are the early activities in the RE phase, in which the involvement of stakeholders to provide information is crucial. This is where crowd-based RE is adapted to improve and simplify the activities. We believe more research should be conducted to explore ways to utilize crowdsourcing techniques and improve the RE process.

As for RQ2, we discover two main challenges in the existing research: stakeholders' coverage and information reliability.

Through the review, we discover that stakeholders' input is crucial to ensure that the information obtained is from genuine sources and reliable. Since it is challenging due to numerous stakeholders, many researchers are exploring ways to involve the various stakeholders and ease the process of getting the information. One of the evolving initiatives is the attempt to provide the stakeholders' analysis tool. Other than that, there are also issues of malicious stakeholders that may respond for their benefit. Therefore, even if we already filter the source of information, which is the stakeholders, at the beginning of the project, it does not guarantee that we can obtain quality requirements. On top of that, by having a diverse set of stakeholders, we may get more relevant and meaningful requirements.

All of the crowd-based RE efforts require the users' involvement to obtain information. iRequire [17], AppEcho [19], FAME [21], SUPERSEDE [25], Classification Model [28], Continuous Requirement Elicitation Methodology [29] and Automated Feature Identification [31] fully rely on the users to provide data to developers. Other than that, Crowdsourced RE Platform for User Story (US) Authoring [32] also fully depends on the users' participation. Besides relying on the crowd input, iRequire, AppEcho, FAME, and Crowdsourced RE Platform for User Story (US) Authoring requires substantial effort to manually perform requirements extraction and refinement. Thus, there are research opportunities to improve the requirements extraction and refinement process after vast information is obtained from the crowd. Furthermore, iRequire [17], AppEcho [19], FAME [21], and Automated Feature Identification [31] can be enhanced in the future as their current capabilities only focus on mobile software.

Besides, there exist efforts that incorporating Artificial Intelligence techniques to process information obtained from the crowd to produce requirements descriptions such as SUPERSEDE [25], Effective Classification Methodology [11], CREeLS [24], Classification Model [28], Continuous Requirement Elicitation Methodology [29] and Automated Feature Identification [31] to handle reliability issues. This is important because the information gathered could be from anybody who gave their responses and feedback. We are well aware that anyone can hook on a web-based software system and mobile apps to give their responses and feedback in this Internet technology era. In their research, M. Bano and D. Zowghi [27] stated that user involvement in software development requires resources and careful management. If the user involvement is not carefully handled, it can cause issues and problems rather than benefits.

There is no single approach that could solve the problem in the traditional RE process. Current crowd-based RE researches span through methods, techniques, tools, and web-based platforms to assist requirements engineering process in many ways while utilizing crowdsourcing benefits. Each of the efforts is unique to solve a specific problem or address the explicit concern in any of the requirements engineering areas.

The efforts are made to take advantage of the crowdsourcing technique and the assistance of the AI technique to obtain quality requirements.

VI. CONCLUSION

In summary, the researchers on crowd-based RE continuously explore the research area and propose new approaches, techniques, and tools to improve the crowd-based RE further. The progressing trend of the research in crowd-based RE proves that this field has more room to improve and to explore. With the rapid growth of technology, particularly the social web and mobile technology, crowd-based RE is becoming more relevant and important to elicit requirements because more people are using the technology to communicate and the software developed to be used by these people must be reliable meeting their needs.

A more comprehensive range of stakeholders can be accessed through crowdsourcing techniques to obtain valuable and meaningful information. Having access to a broader range of stakeholders can provide the breadth of knowledge that leads to quality requirements.

This paper presents the literature on crowd-based RE to complement conventional requirement elicitation techniques to obtain quality requirements. The benefits and advantages of crowd-based RE are worth exploring to strengthen requirement engineering in the future. As for the researchers exploring crowd-based RE, this paper also summarizes the challenges and limitations of crowd-based RE efforts to date.

For future works, it is beneficial to explore the utilization of crowd-based RE to obtain quality software requirements by optimizing the depth and breadth of information at a reduced cost of time and money. We believe crowd-based RE can simplify and improve the RE process to obtain quality software requirements that are later able to produce quality software systems.

ACKNOWLEDGMENT

We are thankful to Universiti Teknikal Malaysia Melaka for funding the publication of this paper through a research grant numbered PJP/2020/FTMK/PP/S01774.

REFERENCES

- [1] J. Howe, "Crowdsourcing: A definition," Retrieved 4 December 2019, from http://www.crowdsourcing.com/cs/2006/06/crowdsourcing_a.html, 2006.
- [2] S.S. Bhatti, X. Gao, and G. Chen, "General framework, opportunities and challenges for crowdsourcing techniques: A Comprehensive survey," *Journal of Systems and Software*, 167, p.110611, 2020.
- [3] U.S. Ghanyani, M. Murad and W. Mahmood, "Crowd-based Requirement Engineering," *International Journal of Education and Management Engineering*, 3, pp.43-53, 2018.
- [4] A. Adepetu, A. A. Khaja, Y. Al Abd, A. Al Zaabi and D. Svetinovic, "Crowdrequire: A requirements engineering crowdsourcing platform," in 2012 AAAI Spring Symposium Series, March 2012.
- [5] J.A. Khan, L. Liu, L. Wen, and R. Ali, "Crowd intelligence in requirements engineering: Current status and future directions," in *International working conference on requirements engineering: Foundation for software quality* (pp. 245-261). Springer, Cham, March 2019.

- [6] A. Ahmad, K. Li, C. Feng, S. M. Asim, A. Yousif and S. Ge, "An empirical study of investigating mobile applications development challenges," *IEEE Access*, 6, pp.17711-17728, 2018.
- [7] A. Inam, A Study of Requirements Engineering Practices Among Software Developers at UUM Information Technology, Ph.D. thesis, Universiti Utara Malaysia. 2015.
- [8] M. Hargrave, "What Is Crowdsourcing," Retrieved 28 January 2020, from <https://www.investopedia.com/terms/c/crowdsourcing.asp>, 2019.
- [9] E.C. Groen, N. Seyff, R. Ali, F. Dalpiaz, J. Doerr, E. Guzman, M. Hosseini, J. Marco, M. Oriol, A. Perini, and M. Stade, "The crowd in requirements engineering: The landscape and challenges," *IEEE Software*, 34(2), 2017, pp.44-52.
- [10] E.C. Groen, J. Doerr, and S. Adam, "Towards crowd-based requirements engineering a research preview," in *International Working Conference on Requirements Engineering: Foundation for Software Quality*, Springer, Cham, March 2015, (pp. 247-253).
- [11] C. Li, L. Huang, J. Ge, B. Luo, and V. Ng, "Automatically classifying user requests in crowdsourcing requirements engineering," *Journal of Systems and Software*, 138, 2018, pp.108-123.
- [12] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," 2007.
- [13] D. Yang, D. Wu, S. Koolmanojwong, A.W. Brown and B.W. Boehm, "Wikiwinwin: A wiki-based system for collaborative requirements negotiation," in *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, IEEE, January 2008, (pp. 24-24).
- [14] S.L. Lim, D. Quercia, and A. Finkelstein, "StakeNet: using social networks to analyze the stakeholders of large-scale software projects," in *2010 ACM/IEEE 32nd International Conference on Software Engineering*, Vol. 1, IEEE, May 2010, pp. 295-304.
- [15] S.L. Lim, D. Quercia, and A. Finkelstein, "StakeSource: harnessing the power of crowdsourcing and social networks in stakeholder analysis," in *2010 ACM/IEEE 32nd International Conference on Software Engineering*, Vol. 2, IEEE, May 2010, pp. 239-242.
- [16] S.L. Lim, D. Damian, and A. Finkelstein, "StakeSource2. 0: using social networks of stakeholders to identify and prioritize requirements", in *Proceedings of the 33rd international conference on Software engineering*, May 2011, pp. 1022-1024.
- [17] N. Seyff, F. Graf and N. Maiden, "End-user requirements blogging with iRequire," in *2010 ACM/IEEE 32nd International Conference on Software Engineering*, Vol. 2, IEEE, May 2010, pp. 285-288.
- [18] S.L. Lim and A. Finkelstein, "StakeRare: using social networks and collaborative filtering for large-scale requirements elicitation," *IEEE transactions on software engineering*, 38(3), 2011, pp.707-735.
- [19] N. Seyff, G. Ollmann and M. Bortenschlager, "AppEcho: a user-driven, in situ feedback approach for mobile platforms and applications," in *Proceedings of the 1st International Conference on Mobile Software Engineering and Systems*, June 2014, pp. 99-108.
- [20] R. Snijders, F. Dalpiaz, S. Brinkkemper, M. Hosseini, R. Ali and A. Ozum, "REfine: A gamified platform for participatory requirements engineering," in *2015 IEEE 1st International Workshop on Crowd-Based Requirements Engineering (CrowdRE)*, IEEE, August 2015, pp. 1-6.
- [21] M. Oriol, M. Stade, F. Fotrousi, S. Nadal, J. Varga, N. Seyff, A. Abello, X. Franch, J. Marco, and O. Schmidt, "FAME: supporting continuous requirements elicitation by combining user feedback and monitoring," in *2018 IEEE 26th International Requirements Engineering Conference (RE)*, IEEE, August 2018, pp. 217-227.
- [22] R. Sharma and A. Sureka, "CRUISE: A platform for crowdsourcing Requirements Elicitation and evolution," in *2017 Tenth International Conference on Contemporary Computing (IC3)*, IEEE, August 2017, pp. 1-7.
- [23] M.Z. Kolpondinos and M. Glinz, "GARUSO: a gamification approach for involving stakeholders outside organizational reach in requirements engineering," *Requirements Engineering*, 2019, pp.1-28.
- [24] N.M. Rizk, A.M. Zaki, E.S. Nasr, and M.H. Gheith, "CREeLS: Crowdsourcing based Requirements Elicitation for eLearning Systems," *International Journal of Ad*, 10(10), 2019.
- [25] A. Perini, "Data-Driven Requirements Engineering. The SUPERSEDE Way", in *Annual International Symposium on Information Management and Big Data*, Springer, Cham, September 2018, pp. 13-18.
- [26] D. Johnson, J. Tizard, D. Damian, K. Blincoe, and T. Clear, "Open CrowdRE Challenges in Software Ecosystems," in *2020 4th International Workshop on Crowd-Based Requirements Engineering (CrowdRE)*, IEEE, November 2020, pp. 1-4.
- [27] M. Bano and D. Zowghi, "Users' involvement in requirements engineering and system success," in *2013 3rd International Workshop on Empirical Requirements Engineering (EmpiRE)*, IEEE, July 2013, pp. 24-31.
- [28] S. Taj, Q. Arain, I. Memon and A. Zubedi, "To apply data mining for classification of crowdsourced software requirements," in *Proceedings of the 2019 8th International Conference on Software and Information Engineering*, April 2019, pp. 42-46.
- [29] A. Alwadin and M. Asharagi, "Crowd, generated data mining for continuous requirement elicitation," *Journal of Advanced Computer Science and Application*, 10, 2019, pp. 45-50.
- [30] J. Wouters, *CrowdRE@ KMar-Case study of the implementation of CrowdRE at the Royal Dutch Marechaussee*, Master thesis, Faculty of Science, Utrecht University, Netherlands. 2020.
- [31] T. Iqbal, N. Seyff and D. Mendez, "Generating requirements out of thin air: Towards automated feature identification for new apps," in *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, IEEE, September 2019, pp. 193-199.
- [32] A. Menkveld, S. Brinkkemper, and F. Dalpiaz, "User story writing in crowd requirements engineering: The case of a web application for sports tournament planning," in *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, IEEE, September 2019, pp. 174-179.
- [33] M. Hosseini, A. Shahri, K. Phalp and R. Ali, "Recommendations on adapting crowdsourcing to problem types," in *2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS)*, IEEE, May 2015, pp. 423-433.
- [34] M. Hosseini, K. Phalp, J. Taylor, and R. Ali, "The four pillars of crowdsourcing: A reference model," in *2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS)*, IEEE, May 2014, pp. 1-12.
- [35] C. Altug, "Crowd-based Requirements Engineering," Retrieved 4 December, 2019, from <https://medium.com/thinkerfox/crowd-based-requirements-engineering-e7e9d044bf71>, 26 September 2019.
- [36] A. Hussain, E.O. Mkpjojogu and F. Mohamad Kamal, "The role of requirements in the success or failure of software projects," *International Review of Management and Marketing*, 6(S7), 2016, pp.306-311.