

# **Crowd Requirement Rating Technique (CrowdReRaT) Model for Crowd Sourcing**

**Adetunji Oluwatofunmi**  
Department of Software  
Engineering,  
Babcock University,  
Ogun State, Nigeria

**Oyenuga Ebenezer**  
Department of Software  
Engineering,  
Babcock University,  
Ogun State, Nigeria

**Otuneme Nzechukwu**  
Department of Computer Science,  
Wesley University,  
Ondo State, Nigeria

## **ABSTRACT**

Requirement Engineering (RE) is an area of software engineering that handles the requirement and elicitation phase which involves the extraction of requirements from business users, clients, and or stakeholders. Producing software that is widely acceptable by prospective users is a function of the accuracy of the requirements gathered during the requirement and elicitation phase of software development. Crowd Requirement Engineering (CrowdRE) is an emerging method that utilizes the power of the crowd kicking out the traditional method for collecting software requirements. The power of a crowd is in its diversity of expertise and talents however, there is still the challenge of managing the crowd, analyzing, and annotating crowd requirements.

This research work aims at developing a multi-level CrowdRE model known as the Crowd Requirement Rating Technique (CrowdReRaT) that enable annotation of requirements by different crowd members at various level. Several models were reviewed systematically to identify their areas of strengths and weaknesses. However, a recommendation for integrating a mini personality survey to know and understand the nature and skillsets of crowd members was suggested.

## **General Terms**

Requirement Engineering, Software Engineering.

## **Keywords**

CrowdRE, Requirement Engineering, Crowd Requirement Rating, Requirement Annotation.

## **1. INTRODUCTION**

Producing software that is widely acceptable by prospective users is a function of the accuracy of the requirements gathered during the requirement and elicitation phase of software development. This phase is as important as any other phase in the Software Development Life Cycle (SDLC) of any software system. Requirement Engineering (RE) is an area of software engineering that handles the requirement and elicitation phase. RE involves the extraction of requirements from business users, clients, and or stakeholders analyzing the validating the requirements [1]. The traditional approach to RE involves having joint meetings with prospective stakeholders to elicit requirements for the proposed system [2]. This approach has from the onset pose several challenges to stakeholders – business owners often time see this process as a form of intrusion to their business thus making them withhold certain information while on the path of the Requirement Engineers, understanding business terminologies, as well as the software domain, is a challenge. Broadly speaking the traditional RE usually involve a limited number of stakeholders representative during the interview which may not completely capture all requirements [1], [3].

In reality, most software products are large with prospective users geographically scattered around the globe which are most time beyond the organization's reach. For such products, having full representatives from all stakeholders does not seem feasible which is also a challenge for traditional RE [4]. Moving on with such a challenge will not guarantee the success of the project as reported by [5] stating that user involvement has a statistically significant association with the success of a project. To overcome the challenges of the traditional RE, Crowd-based Requirement Engineering (CrowdRE) was introduced. CrowdRE is a term used to identify automated or semi-automated approaches to gather and analyze information from a crowd to derive valid user requirements [6]. CrowdRE is an emergent solution to most challenges posed by the traditional RE. Researchers such as [7], [8] have shown that crowdsourcing is a viable, feasible, and promising solution that can be used to collect requirements from varied and diverse users.

However, there is still the challenge of managing the crowd, analyzing, and annotating crowd requirements. Therefore, this research work aims at developing a multi-level CrowdRE model that will be annotated by different crowd members at various levels using a Crowd Requirement Rating Technique (CrowdReRaT) as proposed by the researcher. Several models were reviewed systematically to identify their areas of strengths and weaknesses. This paper is organized as follows: Section 2 presents a brief overview of related works, section 3 discusses existing forms of CrowdRE models, detailed information the proposed model – CrowdReRaT is discussed section 4 while section 5 concludes the research work stating future works and recommendations.

## **2. LITERATURE REVIEW**

Crowd-based RE is an emerging area in Software Engineering. Researchers have shown interest in this area due to its viability. [3] highlighted the promises and challenges of employing the crowd in RE. Others have created techniques or approaches which could be adopted in CrowdRE. [1] presented a crowdsourcing based approach for a German medium-sized company – myERP to help get requirements from non-German customers. The technique proposed was used to extract ERP software requirements. A major pitfall of the presented approach is the fact that there is no way of eliminating duplicate requirements and rating of such requirements. [9] proposed a sequential Crowd RE process where requirements generated from a previous stage is reviewed in the next stage to produce additional requirements. Creative potentials and personality traits of crowd members were gotten through a mini-survey. Eliminating duplicated requirements however remains a challenge. [10] proposed a Crowd- Annotated Feedback Technique (CRAFT) that utilizes the power of the crowd to support richer and more powerful

text mining by enabling the crowd to categorize and annotate feedback through a context menu. This approach helped Requirement Engineers to better identify user requirements in time however, the process of translating (annotating) feedbacks into user requirements is time and effort wasting since feedbacks are gotten after the system has been developed. [11] proposed a gradual approach to crowd-based RE for supporting establishments by involving micro crowds (MCs) with a cohesive and familiar population, the product evolves in each iteration where a new MC can join an already established crowd to enhance the requirements for the next version. This approach as at the time of research was applied to an on-going research project to develop an online social network for academic researchers which was to facilitate discussion and knowledge sharing. However, the research did not factor the motivation of crowd members. [12] came up with a crowd requirement platform known as CrowdREquire alongside with a business model which specifies how RE can help to harness skills available in the crowd. This platform helped in generating system requirements for the proposed

system however, a prototype was not given to the crowd to guide the requirement generation.

Key features and benefits have been highlighted in the above research work. Identifying the crowd, motivating, or rewarding the crowd, providing a working prototype to assist crowd members in the generation of requirements are however identified gaps of the reviewed literature.

### 3. EXISTING MODELS

#### 3.1 CrowdRE Relationship Model

[3] designed a model as shown in figure 1 to show the relationships among all aspects of the CrowdRE. The researcher stated that the CrowdRE relationship model strives to mobilize as many crowd members as possible to communicate and discuss their needs regarding the evolution of existing software products. This model shows the relationships and does not provide a mechanism to annotate crowd requirements or feedbacks.

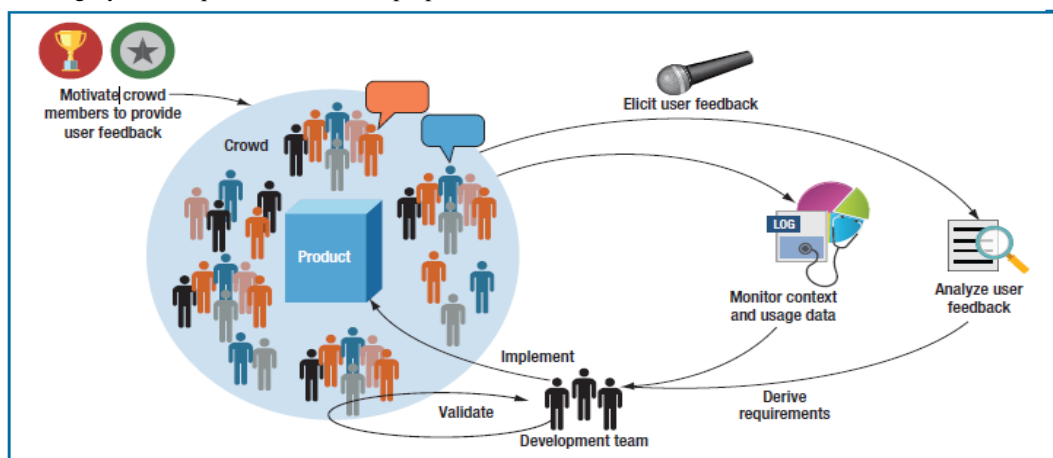


Figure 1: CrowdRE Relationship Model [3].

#### 3.2 CrowdRE Model for OSN

[11] developed a gradual approach model as shown in figure 2 to facilitate a CrowdRE for an Online Social Network (OSN) which was utilized by researchers. Sentiment analysis was utilized in analyzing feedbacks from the crowd. The model aimed at proposing a gradual approach to Crowd-Based RE to mitigate risk in large projects. The approach suggested a gradual building of the crowd in an iterative manner in which

new micro crowds are added at every iteration so that the requirements and user experience will evolve until the system provides complete satisfaction to its users. However, this model fits only a cohesive and familiar population and could take a long while before attaining customer satisfaction, it also does not provide any technique to annotate requirements or feedback from the crowd.



Fig 2: A CrowdRE model for OSN [11]

### 3.3 CRAFT Model

[10] proposed a Crowd Annotated Feedback Technique (CRAFT) model as shown in figure 3 which utilizes crowdsources as a method for harnessing the wisdom of a volunteering crowd to annotate other users' feedback. According to the researchers, the CRAFT model could be used in place of or in addition to automated text mining solutions. In this model, crowd members can annotate any piece of feedback they want at any given time, it also allows a piece of feedback to be annotated several times by several crowd members using predefined categories or new categories as created by the crowd members. This model however does not provide a prototype for crowd members which will stimulate them in generating requirements tailored towards the needs of the proposed software instead crowd members are only allowed to annotate a list of generated feedback.

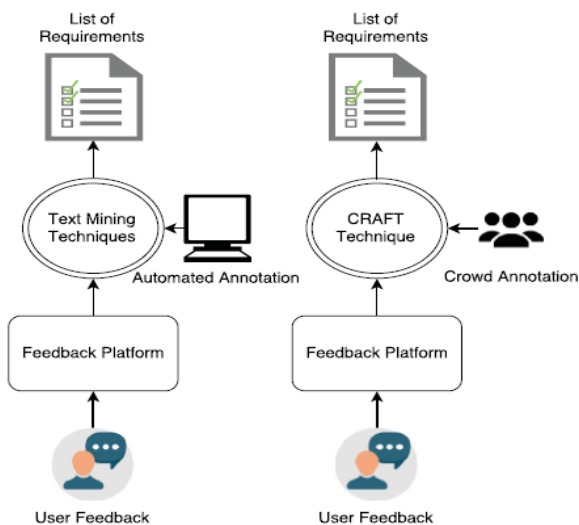


Fig 3: CRAFT model [10]

### 3.4 ERP Requirement Elicitation Model

[1] developed a model as shown in figure 4 to elicit ERP requirements for a German MyERP company from non-German citizens to outshine their competitors in Germany. This model entails some steps which involve identifying the crowd as well as cheaters that may exist among the crowd, the crowd were kept involved in the project by a policy which seeks to reward high-performance workers thus penalizing workers with low performance. Major tasks were categorized to facilitate easy and high-quality requirement elicitation by the crowd. The identified requirements are then prioritized to resolve conflicts. Also, duplicate requirements are eliminated. This proposed model however got crowd members from a single source (LinkedIn) due to the specialization of the product which does not make this model adaptable by any problem domain.

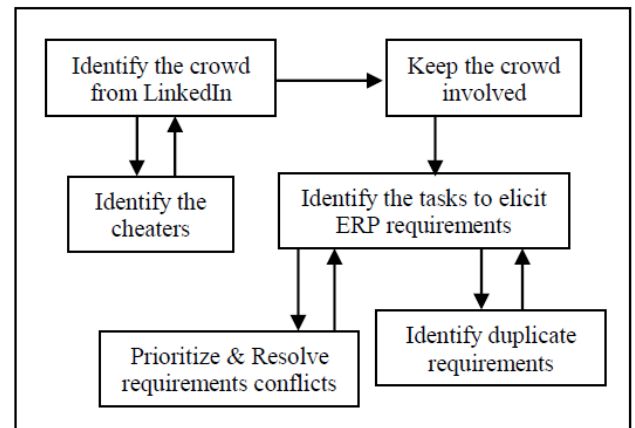


Fig 4: ERP Requirement Elicitation Model [1]

## 4. PROPOSED MODEL - CrowdReRaT

This study seeks to propose a model to combat the challenges associated with managing the crowd as well as annotating the requirements gotten from the crowd. Figure 5 shows the proposed model.

### 4.1 Crowd

Crowd members are prospective users of the software product domiciled in different geographical locations. Crowd members send their feedbacks about the prototype product which is later on annotated into user requirements. [4] identified two types of feedback – pull and push feedback. Pull-feedback takes place when the development team explicitly requests feedback from the crowd while the push-feedback takes place when the crowd initiates the feedback. Feedbacks, in this case, could be linguistic (natural language text, audio messages) and non-linguistic documentation (emojis, images, star ratings).

### 4.2 Motivation

To constantly involve and engage the crowd, it's imperative to motivate (reward) the crowd. As highlighted by [1] crowd members should be motivated if inputs are valued. Motivation could be done digitally through gamification and persuasive technology. Personalities of crowd members should be considered since people could be motivated differently. [3], [13] categorized crowd members with how they will like to be motivated. These highlighted below:

- a) **Privacy-tolerant** crowd members expect acknowledgment in return for their feedback.
- b) **Privacy-fanatical** but generous crowd members are motivated by respect for their privacy.
- c) **Passive and stingy** crowd members are motivated by seeing others' feedback and contributing minimally.
- d) **The loyal and passionate** crowd are motivated and passionate about the software's sustainability and reputation.

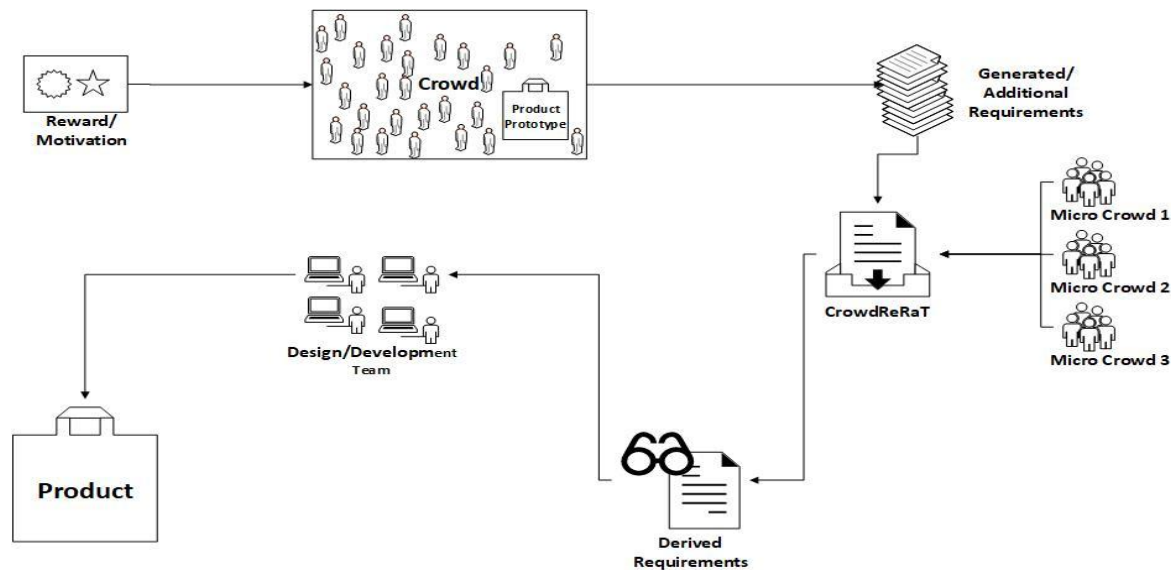


Fig 5: CrowdReRaT Model (Researchers Model)

- e) **Incentive seekers** are about the monetary benefits thereby paying limited attention to feedback and quality.

It is imperative for development firms adopting the CrowdRE technique to identify the motivating factor(s) of crowd members to get the best in them. This can be achieved by carrying out a mini-personality survey on crowd members

### 4.3 Generating Requirements/Additional Requirements

The aim of this phase is for crowd members to generate ideas that turn out to be requirements for the proposed system. To give an idea of the proposed system to the crowd, a miniature (prototype) of the system is presented for study. The prototype is expected to stimulate the generation of ideas. Alongside this stage goes a portal that captures the requirements (ideas) generated. Sample requirements are displayed to the user in a 'user story format' to guide the entry of requirements. Crowd members who come up with creative and distinct ideas will be rewarded specially because creative and distinct ideas will lead to a useful and novel software product as highlighted by [9], [14].

### 4.4 Crowd Requirement Rating Technique

This technique makes use of crowdsourcing as a means of rating the requirements from the crowd. All submitted requirements are captured, from which they are evenly and randomly distributed to crowd members who have participated in the generation process. Crowd members are to rate assigned requirements on a Likert Scale of 1-5 based on clarity, usefulness, and novelty as adapted from [9]. This random rating is done three times by random crowd members – a score for each requirement is generated at each iteration. The average of the three scores is gotten of which requirements with a cumulative score greater than fifty percent (50%) are generated as valid requirements for the system.

### 4.5 Derived Requirements

These are the requirements that have scaled above 50% after cumulation which have been rated based on clarity, usefulness to the domain, and novelty. Duplication of requirements seems to be a minor challenge that can be eliminated through

the help of Requirement Engineers by going through the generated requirements (ideas) and eliminating duplicates.

### 4.6 Implementation

After the elimination of redundant requirements, the Requirement Engineers hand over the requirements to the design and development team who then develop a working and robust system different from the prototype. Depending on the size of the project, the development phase might span months. However, the manner of approach and duration of development is not of the essence in this research work.

### 4.7 Product

The completion of the implementation phase brings about a working software product. This product is subject to testing of several forms (testing is not of the essence in this research work). Once the product has scaled through testing, it would be sent to the initial crowd members for usage to confirm the newly implemented requirements. An opportunity will be given to the crowd members to rate and or give feedback on the newly developed system.

Depending on the stakeholders, the whole steps in the model can be repeated 'n' times until all stakeholders are satisfied.

### 4.8 Challenges and Threats to Model Validity

Crowdsourcing in RE is an emerging trend in the field of Software Engineering. It guarantees stakeholders of getting reliable and cheaper means of generating valid requirements [1], [10]. However, challenges and threats to the validity of the proposed model have been identified as well as the proposed solution. Some challenges are:

- a) **Crowd Identification:** According to [1] the quality of the gathered requirements depend on the crowd. It is therefore of great essence to identify crowd members who can make a significant contribution. To overcome the challenge of crowd identification, the researcher proposes getting suited participants from various social media platforms such as LinkedIn (recommended), Facebook, Instagram, and the likes. Conducting a mini-survey to identify skill sets and personalities will help suppress this challenge.

- b) **Engagement/Retention of Crowd Members:** This has been identified as one of the biggest challenges in crowdsourcing [15]. The quality of the collected requirements is greatly dependent on the crowd's involvement and how they are valued [1]. The problem of crowd retention and engagement can be solved by identifying the type of reward that suits each crowd member as identified in section 4.2. It is therefore important in making provision for recognizing the performance of crowd members.
- c) **Generating Reliable Requirements:** Since the crowdsourcing technique involves the use of a crowd in which there are no formal contracts or employment, there is the tendency of getting some requirements that are not suitable for the problem domain. The CrowdReRaT model is proposed to solve this problem. First the crowd members are

given a prototype of the proposed software which is expected to stimulate the ideas needed for a complete version of the proposed system. Secondly, a sample requirement is presented to crowd members in a 'user story' format to show the crowd how to translate their ideas into functional requirements.

- d) **Duplication of Requirements:** Employing the crowd in coming up with ideas (requirements) means employing people who are working independently and scattered geographically. These most times bring about the tendency of duplication in the generated requirements. Duplicates of requirements should be eliminated as soon as they are identified by the Requirement Engineers.

Table 1 shows the unique features of each model discussed above.

**Table 1. Feature Comparison of CrowdRE Models**

Features	CrowdRE Relationship Model	Online Social Network OSN Model	CRAFT Model	ERP Requirement Elicitation Model	CrowdReRaT Model
Organized mobilization of crowd	✓	✓	✓	✓	✓
Crowd identification	–	–	–	✓	✓
Crowd motivation and reward	✓	–	–	✓	✓
Identification of cheating crowd members	–	–	–	✓	–
Specialized crowds	–	✓	–	✓	✓
Generalized crowds	✓	–	✓	–	✓
Prototype requirements	–	–	–	–	✓
Elicitation of requirements	✓	✓	✓	✓	✓
Annotation of requirements	✓	–	✓	–	✓
Elimination of duplicate requirements	–	–	–	✓	✓
Risk mitigation	–	✓	–	–	✓
Requirement rating	–	✓	–	–	✓
Evaluation of products by initial crowd members	✓	–	–	–	✓

## 5. CONCLUSION

The traditional approach to RE is complex and challenging for a software system that requires global acceptance. In this paper, a CrowdReRaT model has been proposed which could be adopted by any problem domain of software development. Several threats and challenges were identified as well as proposed solutions. However, as future work, a mini personality survey to know, categorize, and identify the skill set of each member of the crowd will be integrated.

## 6. REFERENCES

- [1] P. K. Srivastava and R. Sharma, "Crowdsourcing to elicit requirements for MyERP application," *1st Int. Work. Crowd-Based Requir. Eng. CrowdRE 2015 - Proc.*, pp. 31–35, 2015.
- [2] I. Sommerville, *Software Engineering*, 9th ed. Pearson. India., 2011.
- [3] E. C. Groen *et al.*, "The Crowd in Requirements Engineering: The Landscape and Challenges," *IEEE Softw.*, vol. 34, no. 2, pp. 44–52, 2017.
- [4] W. Maalej, H.-J. Happel, and A. Rashid, "When users become collaborators," in *In Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and app*, 2009, p. 981.
- [5] S. Kujala, M. Kauppinen, L. Lehtola, and T. Kojo, "The role of user involvement in requirements quality and project success," in *Proceedings of the IEEE International Conference on Requirements Engineering*, 2005, pp. 75–84.
- [6] E. C. Groen, "Crowd Out the Competition: Gaining Market Advantage through Crowd-Based Requirements Engineering," *1st Int. Work. Crowd-Based Requir. Eng. CrowdRE 2015 - Proc.*, pp. 13–18, 2015.

- [7] K. Mao, L. Capra, M. Harman, and Y. Jia, "A survey of the use of crowdsourcing in software engineering," *J. Syst. Softw.*, vol. 126, pp. 57–84, 2017.
- [8] K. P. J. T. and R. A. M. Hosseini, "Toward Crowdsourcing for Requirements Elicitation: Results from Expert Survey," in *Proceedings of the Empirical Track of REFSQ*, 2014.
- [9] P. K. Murukannaiah, N. Ajmeri, and M. P. Singh, "Acquiring Creative Requirements from the Crowd: Understanding the Influences of Personality and Creative Potential in Crowd RE," *Proc. - 2016 IEEE 24th Int. Requir. Eng. Conf. RE 2016*, pp. 176–185, 2016.
- [10] M. Hosseini, E. C. Groen, A. Shahri, and R. Ali, "CRAFT: A crowd-annotated feedback technique," *Proc. - 2017 IEEE 25th Int. Requir. Eng. Conf. Work. REW 2017*, pp. 170–175, 2017.
- [11] M. Levy, I. Hadar, and D. Te'eni, "A gradual approach to crowd-based requirements engineering: The case of conference online social networks," *1st Int. Work. Crowd-Based Requir. Eng. CrowdRE 2015 - Proc.*, pp. 25–30, 2015.
- [12] A. Adepetu, K. A. Ahmed, Y. Al Abd, A. Al Zaabi, and D. Svetinovic, "CrowdREquire: A requirements engineering crowdsourcing platform," *AAAI Spring Symp. - Tech. Rep.*, vol. SS-12-06, no. Goodin 2005, pp. 2–7, 2012.
- [13] M. Almaliki, C. Ncube, and R. Ali, "Adaptive software-based Feedback Acquisition: A Persona-based design," in *IEEE 9th International Conference on Research Challenges in Information Science (RCIS)*, 2015, pp. 100–111.
- [14] M. A. Runco and G. J. Jaeger, "The Standard Definition of Creativity," *Creat. Res. J.*, vol. 24, no. 1, pp. 92–96, 2012.
- [15] C. Puah, A. Z. Abu Bakar, and C. W. Ching, "Strategies for community based crowdsourcing," in *2011 International Conference on Research and Innovation in Information Systems, ICRIS'11*, 2011, pp. 1–4.