ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ENHANCEMENTS FOR CROWDSOURCED REQUIREMENTS ENGINEERING

[1] SULTAN ALYAHYA, [2] WEJDAN ALOHALI, [3] SAMEERAH AL-BALHARETH

[1,2] Department of Information Systems, College of Computer and Information Science, King Saud University, Saudi Arabia

[3] Department of Computer Science, College of Education-Jubail, Imam Abdulrahman bin Faisal University, Saudi Arabia

E-mail: [1]sualyahya@ksu.edu.sa, [2]435920022@student.ksu.edu.sa, [3]smbalhareth@iau.edu.sa

ABSTRACT

The use of crowdsourcing in software requirements engineering has become common nowadays. The web-based services offered through Crowdsourced Requirements Engineering (CRE) platforms support customers in finding timely and accurate requirements specification for their proposed tasks and projects. Currently, the literature does not have a critical assessment of the key activities involved in the CRE platforms. In this paper, we review the process used in the CRE platforms including identifying the workflow used in managing the process. Then we made a step further in the direction of improving the current platforms; the review is used to identify a set of limitations in the current process and has led to propose enhancements. These enhancements are evaluated using two techniques: questionnaire and workshop. The questionnaire shows that the enhancements are sound and practical to be added to CRE platforms. In addition, the evaluation through conducting a workshop showed that the participants were satisfied with the enhancements but asked for further modifications.

Keywords: *Crowdsourced Requirements Engineering, Crowdsourcing, Requirements Engineering, Enhancements, Platforms.*

1. INTRODUCTION

Nowadays with the spread of the Internet, many tasks are fulfilled by the crowd. Software organizations have begun to seek the use of crowdsourcing web-based services within software development projects. Crowdsourcing can be used for different purposes including requirements engineering. Requirements engineering is part of software engineering that defines the functions that the system will provide and its constraints. Poor requirements engineering and misunderstanding of the software requirements are major reasons for project failure [4].

To develop high quality project requirements, a possible solution is to use crowdsourcing rather than traditional in-house approach as more wide range of people can participate in gathering system requirements. This change from traditional to crowdsourcing reduces the workload on analysts and enables them to have more time available for other tasks/projects. According to Srivastava and Sharma [1], traditional methods of requirement gathering are more costly and time-consuming for a projected major software system with diverse users. Diversity can be an important aspect in crowdsourcing since it can help provide more creative and relevant requirements. Adepetu et al. [2] mention the significance of diversity on requirements gathering and state that the crowd's power is on its talent diversity, together with the availability of expert within the crowd. In addition, Snijders et al. [3], state that CRE leads to high satisfaction of users and better quality of requirements.

Although there are currently many crowdsourced requirements engineering (CRE) platforms, the capabilities of these platforms do not provide sufficient support to the process of requirements engineering in software projects. Many issues on the level of quality, consistency, completeness, communication, coordination and planning have been reported in the literature [2] [10]. Consequently, requirements engineers and end users find it challenging and time consuming to establish and maintain their projects' requirements, affecting the overall delivery and execution [3]. Nevertheless, the literature does not have a critical assessment of the key activities involved in those platforms.

The focus of this research is to provide a better understanding of the requirements engineering process within the current CRE platforms. Then, it makes a step further in the direction of improving the current platforms; the review is used to identify a set of limitations in the current process and then to propose software enhancements.

The remainder of this paper is organized as follows: section 2 provides some background information, section 3 discusses the related work and section 4 describes the current CRE platforms' workflow. Observations and limitations on the current process are discussed in section 5 while the enhancements are proposed in section 6. Finally, the evaluation is presented in section 7 and the discussion and conclusions are provided in section 8.

2. BACKGROUND

In this section, we provide a short background needed to understand this research. In particular, we will provide a short description about requirements engineering and crowdsourcing.

2.1 Requirements Engineering

Requirements Engineering (RE) is the branch of software engineering that that focuses on the processes of handling and management of requirements in any software development effort [4] [5]. It focuses on the process of generating requirements through the understanding of customers' needs.

Maalej et al. [6] define five activities for requirement engineering process as listed below:

- Requirements elicitation: it is the process through which user's requirements and constraints are discovered and understood.
- Requirements analysis: It is the process of reviewing and refining the requirements and constraints of customer.
- Requirements specification: it is the process where user's needs and constraints are clearly documented and defined.
- Requirements verification: it is the process where system requirements are checked for correctness, consistency, clarity, and completeness.
- Requirements management: it is the process where stakeholders schedule, coordinate, and document the RE activities.

2.2 Crowdsourcing

Crowdsourcing has the potential as a technique for greater user involvement [7] [8]. Howe [8] defines crowdsourcing as the act of a firm or company taking a role or function once performed by a group of workers and outsourcing it to a large network of workers/people using an open call.

Crowdsourcing is one of the major internet-based services; the main characteristics of crowdsourcing are its highly collaborative environment, its ability to explore new ideas, its very fast process and, most importantly, its significantly low production cost [9].

Hosseini et al. [10] have classified the main parts of crowdsourcing into:

- The crowd: individuals or group of people participating in activities connected to crowdsourcing.
- The crowdsourcer: This is the entity that utilizes the crowdsourcing concept to complete a task.
- The crowdsourcing task: This is the activity being completed using the crowdsourcing concept.
- The crowdsourcing platform: This is the system or setting where the crowdsourcing task is accomplished.

3. LITERATURE REVIEW

This section points out the most relevant research on the use of crowdsourcing for requirements engineering field.

Since large percentage of projects fails due to wrong identification of stakeholders, Lim et al. [11] created a 'StakeNet Methodology' for the identification and analysis of stakeholders. StakeNet requires to build a network with the nodes representing stakeholders, and prioritizes stakeholders using several social network approaches.

StakeRare method is also used in [12] for requirements elicitation & prioritization as a complementary to StakeNet. This method is founded on collaborative filtering and social network analysis methods. StakeRare has the capability to elicit a full set of requirements that can be precisely prioritized.

Hosseini et al. [13] present a study that confirms the relationship between crowdsourcing features and the quality of requirements. In the findings, the authors indicate that 47.06% are in agreement that using a large crowd contributes to greater accuracy in requirements, and there is difficulty in coordinating and organizing large crowds for requirements elicitation.

Identifying the best-suited participants with domain knowledge is discussed by Wang et al. [14]. They focus on defeating the problem of stakeholder recruitment. They proposed a recruitment framework based on spatiotemporal availability. In their simulation, they demonstrated the practicality and feasibility of the proposed method.

Travis and Florian [15] conducted three experiments on privacy policy documents. Requirements extraction experimental results show a 16% growth and a 60% cost reduction.

Nixon et al. [16] acknowledge that poor requirements engineering is a major reason on project failure in software development and crowdsourcing can be selected to solve the inefficiencies of IT project prioritization.

To reduce risks in crowd-based requirements engineering projects which include lower engagement of individuals of crowd, Levy et al. [17] propose steadily building crowd in an iterative way by placing new micro-crowds (MC) into the new iterations.

Furthermore, requirements engineering within crowdsourcing is used for mobile applications. Norbert and Florian [18] propose iRequire tool which is a mobile elicitation tool that enables the users of the social software to record the needs in its original place, this tool automatically apprehend contextual information. iRequire can update the requirements negotiation through aiding stakeholders in effectively understanding the actual needs and give essential information that can update developers regarding the accessibility of GPS positioning data.

To obtain the user's feedback and consideration regarding requirements engineering, Groen, et al [19] establish Crowd-based Requirements Engineering method that gathers feedback via social collaboration and direct interactions and also through using mining approaches. In addition, they have used the user feedback gathered from crowd to derive the clues, bug reports, ideas, wishes, and needs regarding the trends with the aim of anticipating new innovations.

Furthermore, Walid et al. [20] have analyzed different methods, tools, including techniques for feedback analytics that can enhance the feedback quality and help developers in comprehending the feedback. These tools manage the vast number of user feedbacks by categorizing, sorting, and summarizing user comments. Moreover, requirement managers and analysts can be assigned these issues to enable them to discuss and point out some requirements for the future releases.

Mao et al. [21] provide a detailed survey of the use of crowdsourcing in software development activities. This involves looking at crowdsourcing in requirements analysis. This review has brought to light a new trend where the theories, application and practice in this area are being continuously published. They state that coordination and management between participants in CRE process is one of the main limitations. Lacking notifications and awareness between crowd team may cause running the work inefficiently and provide unmanaged crowd team workload.

Although there are many research works about the use of crowdsourcing in requirements engineering, there is no available research assessing the main activities in CRE platforms.

4. REVIEW OF CRE PLATFORMS

4.1 Platforms Covered

The platforms are selected based on popularity and support for the crowdsourced RE as mentioned in a detailed survey of the use of crowdsourcing in software development activities [21]. These platforms are:

- Stake-Source [22][35]: It is a web-based platform for requirements elicitation. It aims to recommend other stakeholders to help in requirements elicitation process.
- StakeSource 2.0 [11]: It is a web-based platform for requirements identification and prioritization. It aims to identify and prioritize stakeholders and their requirements.
- Crowd Require [2]: It is a web-based conceptualized crowdsourcing platform for requirements engineering. It aims to design a crowdsourcing business model and market strategy for RE.
- Winbook [23]: It is web-based platform for requirements elicitation, and negotiation.
- iThink [24]: It is a web-based platform for requirement elicitation which support gamification.

- Requirements Bazaar [25]: It is a web-based platform for requirements elicitation, negotiation, prioritization. It aims to facilitate the communication and negotiation process between communities and service provider.
- REfine [26]: It is a web-based platform for requirements elicitation and refinement. It aims to provide participation incentives via gamification.

4.2 Current Workflow

CRE platforms use a common workflow to manage the process of crowdsourcing the requirements. Figure 1 presented below shows this workflow with the actors described as follows:

- **Cowdsourcer** is any person, company or organization that would like to use the crowd to elicit their requirements. They are responsible for initiating and creating the project and ending the whole process as they have the final decision to approve the delivered requirement specifications.
- **Crowd Manager** is the person responsible for monitoring the system activities and reviewing the submitted project. He is also responsible for managing any possible conflicts in requirements.
- **System (Platform)** provides a communication platform to connect requirement-engineering experts with the companies that need their services.
- **Crowd** is a group of those who register on the system and are willing to participate by submitting a requirements solution to any of the projects offered.

The main activities involved in the workflow are described below.

- Submitting Project

  The crowdsourcer creates the project and enters a project description, title, scope, and recommends primary stakeholders (crowd) by including their details such as name, email, role and etc.

- Reviewing Project by Crowd Manager

  The crowd manager will be notified by the system in case of any posting of new task. After the creation of a project, the crowd manger reviews the submitted project to categorize the requirements.

- Categorizing Project

  The crowd manager categorizes the project based on actors' roles by using project description, area of specialization and the nature of projects for grouping defined requirements.

- Inviting Crowd

  The system sends an invitation to each registered crowd member by using the project details to inform the crowd about the project. On the other hand, crowd can register by subscribing to event notifications. The event notification updates the crowd whenever a new project is submitted or if the existing requirements have been updated.

- Reviewing New Project by Crowd

  The crowd's member has the ability to review new projects and has a choice to accept or reject the invitation to participate due to the crowd's availability and interests.

- Negotiating Requirements

  Negotiation and discussion among crowd and crowdsourcer could take place. If the crowd member is interested, then the crowd commits to participate in requirements elicitation.

- Recommending Crowd

  After a negotiation is completed, the crowd can decline the participation in the project if they were preoccupied or have no interest in the submitted project. Yet, the crowd can recommend other crowd members to participate by entering their names, roles and email addresses. After that, the system can send invitations with project details to the recommended crowd.

- Submitting Project Requirements

  After selecting a project, the crowd can provide a list of new requirements.

- Providing Feedback by Crowdsourcer

  The crowdsourcer has the ability to review the submitted requirements. Based on the crowdsourcer's needs, requirements can be either accepted or rejected and sent back with crowdsourcer's feedback.

- Reviewing Required/Existing Project

  The crowd's member has the ability to review existing requirements and has a choice to accept or reject the invitation to provide feedback.

- Providing Feedback by Crowd

  The crowd can express their opinion on a given requirements in four different ways. These are comments to existing requirements, voting on requirements via agree/disagree options, branching, and rating a requirement with stars (i.e. giving five-star as rate means very important requirements, one-star means it is not important). However, the crowd cannot express opinions about their own requirements.

- Prioritizing Requirements

  A relative priority assigned to each requirement, is estimated by calculating the sum of the ratings of that requirement then finding the average and show a prioritized list.

- Obtaining Result

  The system displays the requirements in a prioritized list along their ratings and the crowd members who rated them.

- Managing Conflict

  The crowd manager should pay more attention to a requirement if many crowd members are in conflict. The manager, based on previous experience, knowledge domain and role of crowd members, can make the final decision and this requirement may be marked as agreed or disagreed.

- Choosing Appropriate Project Requirements

  The crowdsourcer has the final decision to approve and choose one of the delivered requirement specifications.

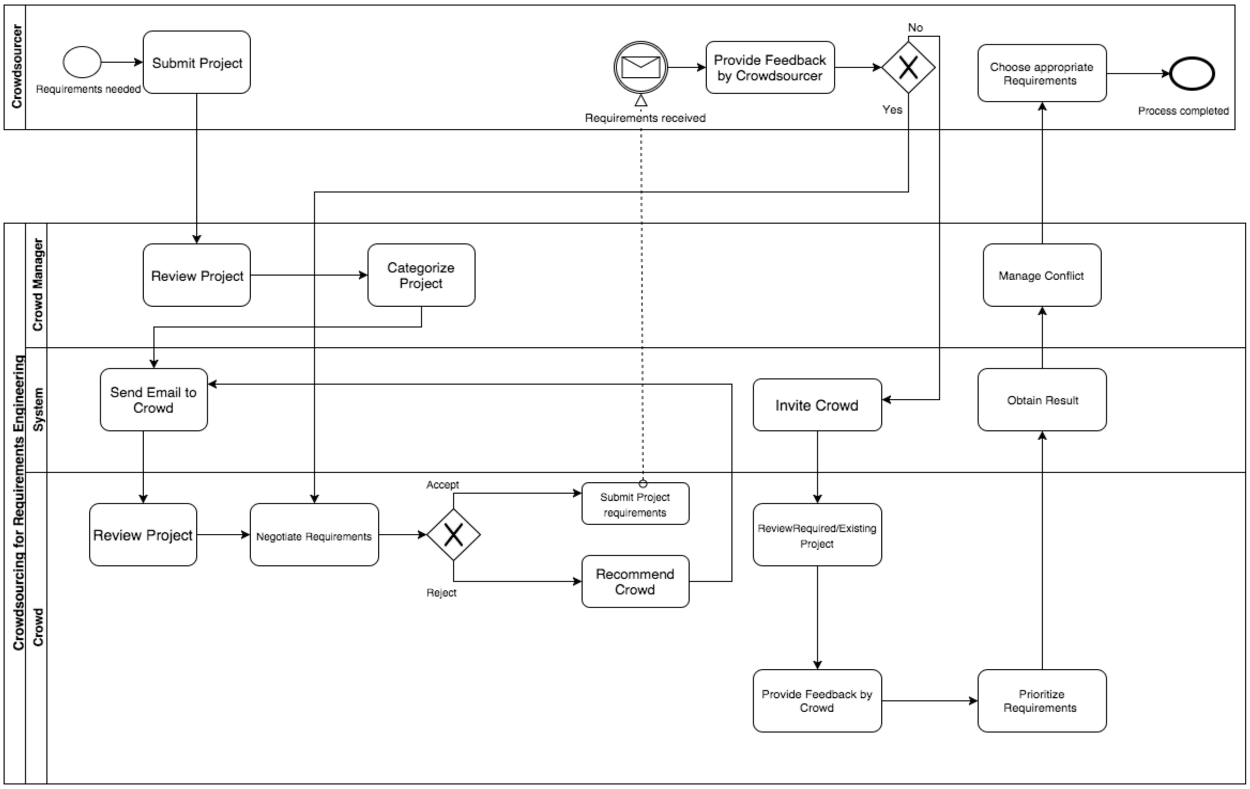

Figure 1: Current workflow of CRE platforms.

Table 1 below shows the supported and unsupported activities in each of the CRE platforms.

Table 1: Summary of current platforms along with supported and unsupported activities
Key: ✓Activity is supported. ✗Activity is not supported.

Platform Activity	Staksource	Stakesource 2.0	CrowdREnquire	WinBook	iThink	Requirements Bazaar	REfine
Submit Project	✓	✓	✓	✓	✓	✓	✓
Review Project by Crowd Manager	✕	✕	✕	✕	✓	✕	✕
Categorize Project	✕	✕	✕	✕	✓	✕	✕
Invite Crowd	✓	✓	✕	✕	✕	✓	✕
Review New Project by Crowd	✕	✓	✓	✓	✓	✓	✓
Negotiate Requirements	✕	✕	✓	✓	✕	✓	✕
Submit Project Requirements	✕	✓	✓	✓	✓	✓	✓
Provide Feedback by Crowdsourcer	✕	✕	✓	✓	✕	✓	✓
Review Required/ Existing Project	✕	✓	✓	✓	✓	✓	✓
Provide Feedback by Crowd	✕	✕	✓	✓	✓	✓	✓
Priotrize Requirements	✕	✓	✓	✕	✕	✓	✕
Obtain Result	✕	✓	✕	✓	✕	✕	✕
Manage Conflict	✓	✕	✕	✓	✕	✕	✕
Choose Appropriate Project Requirements	✕	✓	✓	✓	✓	✓	✓

5. LIMITATIONS

We have identified a set of limitations based on our analysis of the CRE platforms. These are discussed below

i. Submitting Project

When crowdsourcer submits a project, the project description, scope and title are determined while some important data such as project duration and initial project's preferences are not. This can be a point of fail. Let us assume that one university needs to have a blackboard system for its students and teachers. The university (crowdsourcer) has decided to use CRE platforms to collect potential requirements.

The process starts by submitting the project and filling the required fields, which are project description, scope and title. The submitted project will be handled within an open duration which means the project will not be closed and it is handled by the crowd who may have no interests in education field. Therefore, lacking the mentioned missing fields may have an impact on the project quality and project time.

ii. Reviewing Project

Through the current CRE platforms, the crowd manager is able to review the project in order to categorize the project requirements only. The current 'reviewing project' activity does not support evaluating projects based on project size, complexity, and duration. Lacking evaluating and reviewing the project in a proper way can have an impact on the submitted solution.

iii. Crowd's Invitation and Participation

The system sends an invitation to crowd without checking the competence and preferences of the crowd. For example, the submitted system may be related to education field and the system sends an email to the crowd who has interest in other domains such as health or finance.

Furthermore, after accepting the project by the crowd, the crowd manager is not aware about who will participate in the project and he is not notified to ensure that the participated crowd is qualified (e.g. time zone, work hours availability). This limitation stresses the need to provide coordination mechanisms supporting managing the activities involved in CRE.

iv. Submitting Project Requirements

The crowd can submit their solutions to crowdsourcer without involving and notifying the crowd manager. In fact, the submitted solutions need to be reviewed to assure not providing overlapped, duplicated or inconsistent requirements.

v. Providing Feedback by Crowdsourcer

The crowdsourcer has an opportunity to provide their feedback about the submitted result. However, there is no deadline to receive the feedback. This can cause a delay in closing the project.

6. PROPOSED ENHANCEMENTS

In this section, processes that shall help overcome the observations and limitations mentioned in in the previous section will be presented.

A. Developing Crowd's Profile

To overcome limitation (i), every crowd member needs to set their own personal profile. In order to complete the profile creation process, the crowd needs to identify several competences, specify their time zone, years of experience, work hours availability, personal time constraints (e.g. working on weekends), the crowd's work role and status (i.e. available, unavailable). After the creation process is completed, the profile can be automatically categorized within the system based on the crowd's competence. Ensuring the development of a complete profile is a contributing factor into the project success as it could influence the task assignment process and make it faster and more accurate as well as helping in allocating the crowd properly. Figure 2 presents the developed crowd's profile process workflow.

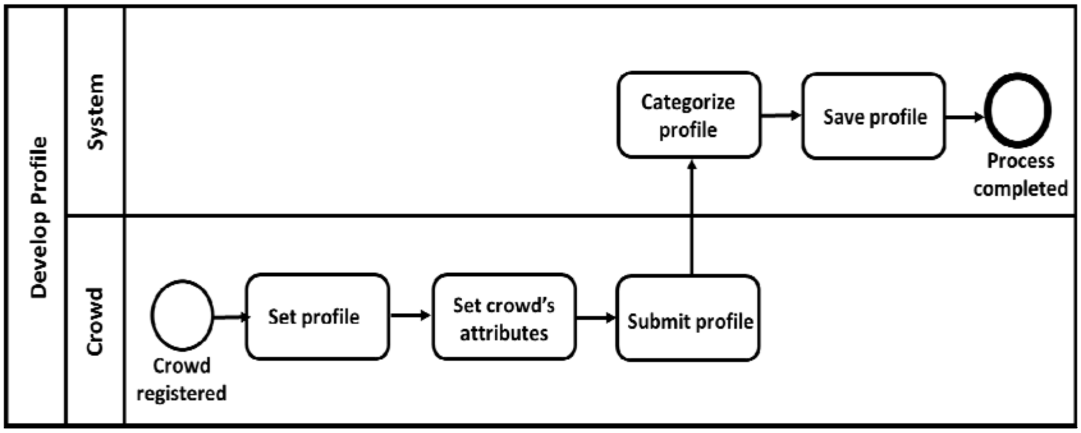

Figure 2: Developing crowd's profile and Auto-categorizing it workflow.

B. Auto-Categorizing Project

To overcome limitation (ii), the project can be categorized automatically in order to help reduce the time and effort of the manager. One of the project categorization factors that can be considered is the project preference, which is determined by the crowdsourcer when submitting the project. There are two reasons that adds importance to this factor: first, several projects could share similar preferences which makes it easier to categorize them together. Second, the crowd knowledge, competence and capabilities are crucial in building teams and to allocate/assign the correct project to the correct crowd. Therefore, project preferences are important to facilitate the crowd profile mapping process, which depends on the crowd competence and project preferences. Indeed, the project preferences factor can reduce the project cycle time and remove the manager's unnecessary work so the project can run faster.

This process shall start when the project preferences are submitted by the crowdsourcer. Then, the system shall map the project preferences with the proper categorization. Auto-categorization for the project will be applied based on the actors' roles. Finally, the system shall retrieve the categorization and notify the crowd manager. Figure 3 shows the proposed categorization project process workflow.

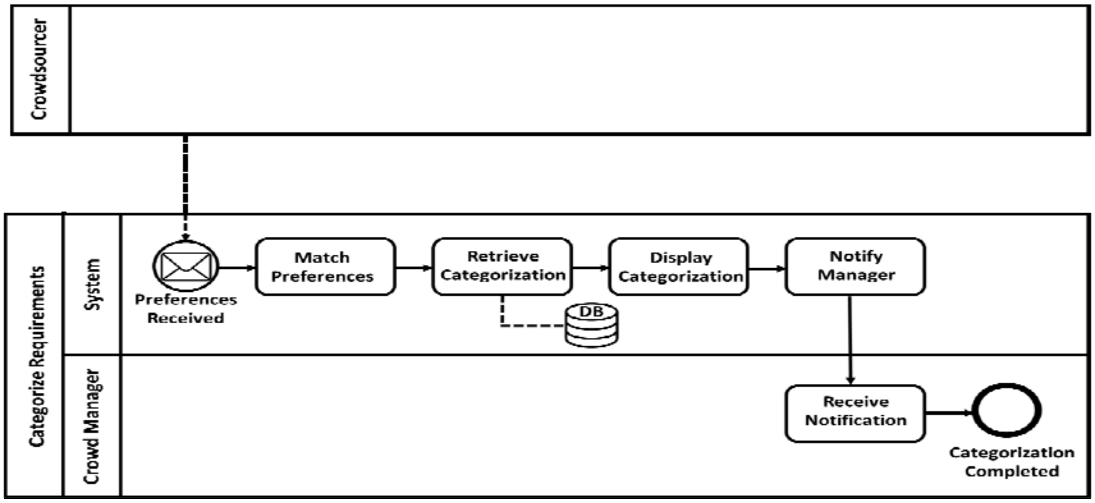

Figure 3. ` Auto-categorizing project workflow.

C. Creating Project Plan

To overcome limitation (iii), it is important that the crowd manager be able to set a clear project plan. A possible process can include:

a. allocating the right crowd member to the right task.
b. Specifying appropriate Team Size
c. Developing appropriate project schedule

It is essential that the crowd manager ensures allocating the right crowd to the right task and to have the optimal team size. Thus, the crowd manager shall develop a clear time plan for assigning tasks to the crowd team. The process starts by auto-developing the project schedule. The crowd manager shall review the project scope and specify the project team size. Then, the project manager will decide to break down the project or not, based on the project complexity. If so, for each task the crowd manager needs to estimate the following: the task duration and the optimal number of team members needed to complete the task. Auto-mapping the crowd competence and the task preferences is applied in order to send an invitation to the proper crowd. Figure 4 shows the process workflow.

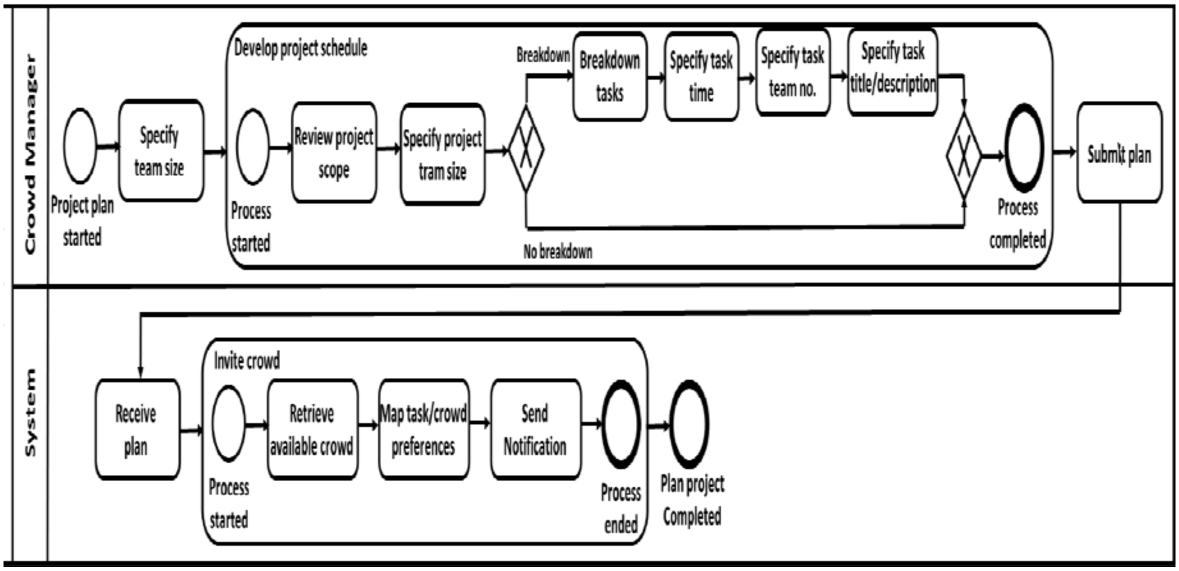

Figure 4: *Creating project plan workflow.*

D. Managing Requirements Overlapping and Duplicates

To overcome limitation (iv), this enhancement assures the submitted requirements quality (i.e. correctness, completeness and fulfilling the crowdsourcer's needs). When the crowd submit the requirements, the system needs to notify the crowd manager to check the requirements' quality. After that, the project result will be sent to crowdsourcer. Figure 5 shows the workflow.

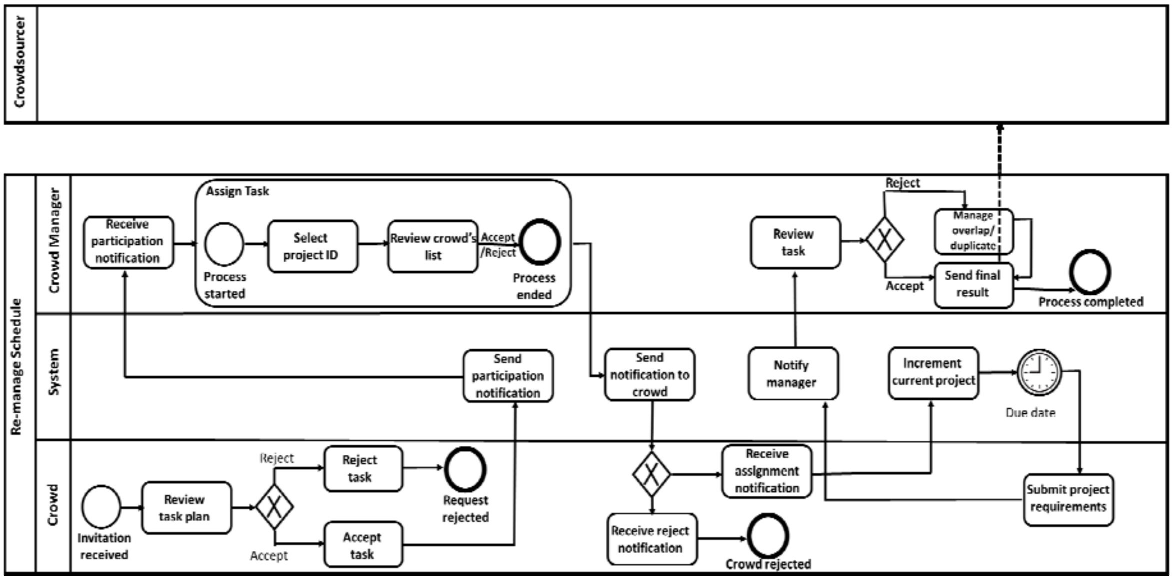

Figure 5: Managing overlapping workflow.

E. Managing Crowdsourcer's Feedback

To overcome limitation (v), the crowd manager shall create a clear project plan and ultimately close the project after receiving the final confirmation from the crowdsourcer.

Consequently, the system needs to be configured to start counting down from the final result submission date till the feedback due date. If the feedback exceeds a determined period of time (e.g. twenty-four hours), then the system shall automatically close the project. Otherwise, the crowdsourcer's feedback will be sent to the crowd manager in order to evaluate it. If it is accepted, the crowd manager shall create a feedback plan, which includes the task, and estimated date, which would start the feedback work cycle. After that, the system shall close the project, increment the total completed projects for crowd and decrement the total current assigned projects. Figure 6 shows the process workflow.

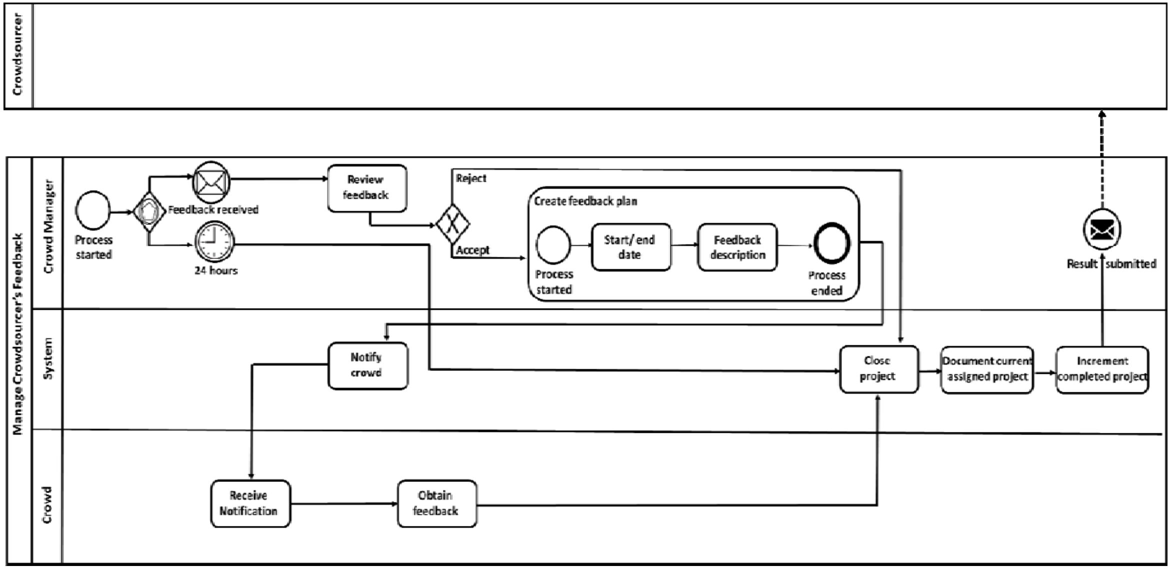

Figure 6: Managing crowdsourcer's feedback workflow.

7. EVALUATION

The evaluation of the enhancements is conducted through two approaches: questionnaire and workshop. These are discussed below.

7.1 Questionnaire

Online questionnaire is created in order to evaluate our proposed enhancements. The questionnaire

The questionnaire is distributed through the email and social media. The questionnaire includes some attached videos of the implemented processes to demonstrate the workflow.

The questionnaire consisted of 6 questions. A total of 21 respondents took part in our questionnaire. In this section, we analyze the results of the questionnaire for each enhancement.

The questionnaire questions are formulated as follows:

Q1. What is your work role when using crowdsourcing platforms?

The initial question is to indicate the primary role of the respondents. Our respondents have different roles when using crowdsourcing for RE tools. The majority of respondents are Crowd with 38.1% followed by Crowd manager with 28.6% as shown in Figure 7.

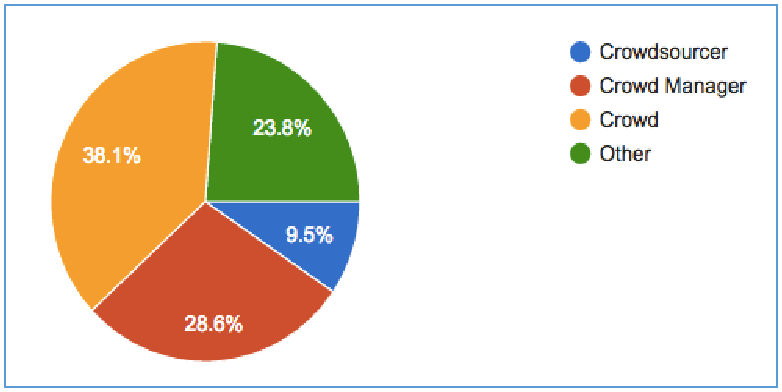

Figure 7: Roles of Respondents.

Q2. Developing Crowd's Profile

Figure 8 shows that the majority of respondents either agree or strongly agree with the importance of having complete and well-developed profile for the crowd. As mentioned earlier, developing complete profile will have an impact on the task assignment process since it would be more accurate and faster. It will help in allocating the crowd properly as well.

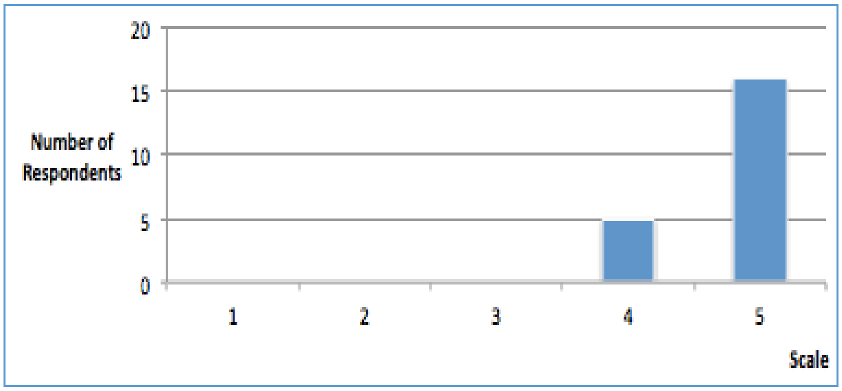

Figure 8: Evaluation of Developing Crowd's Profile.

Q3. Auto-Categorizing Project

Almost all the respondents agree and confirm the importance of auto-categorizing the projects based on the project preference (Figure 9).

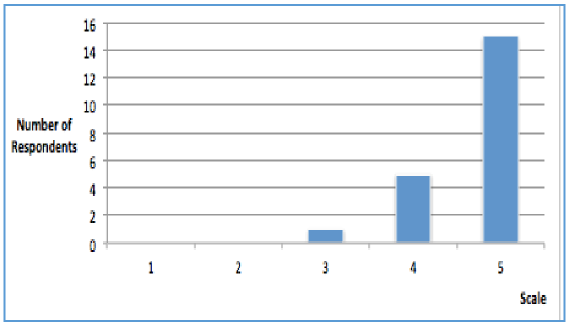

Figure 9: Evaluation of Auto-Categorizing Project.

Q4. Creating Project Plan

It is clear from Figure 10 all the respondents agree with building a team with competence and complementary skills and auto-developing project schedule.

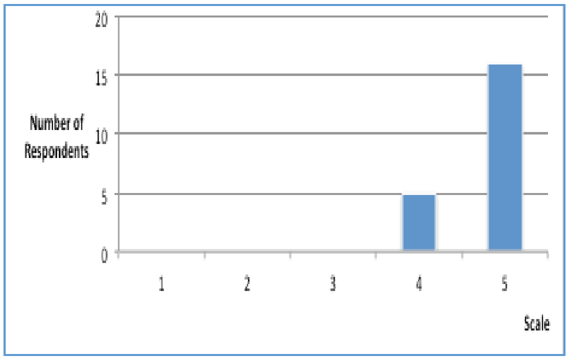

Figure 10: Evaluation of Creating Project Plan.

Q5. Managing Requirements Overlapping and Duplicates

It is suggested that the crowd manager is automatically notified when the task is done, to assure that the requirements are complete and correct and fulfill the crowdsourcer's needs. The majority of the respondents agree with this enhancement (Figure 11).

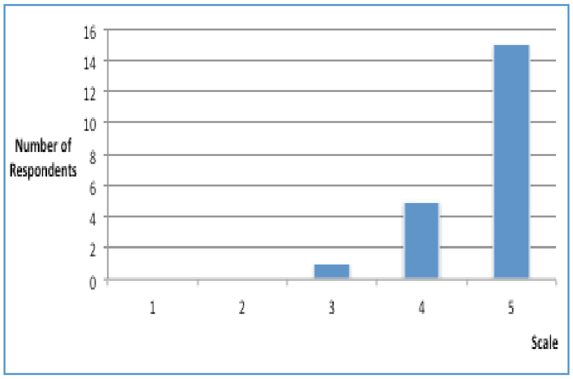

Figure 11: Evaluation of Managing Overlapping.

Q6. Managing Crowdsourcer's Feedback

Figure 12 shows that more than half of the respondents agree or strongly agree with the importance of managing the crowdsourcer's feedback.

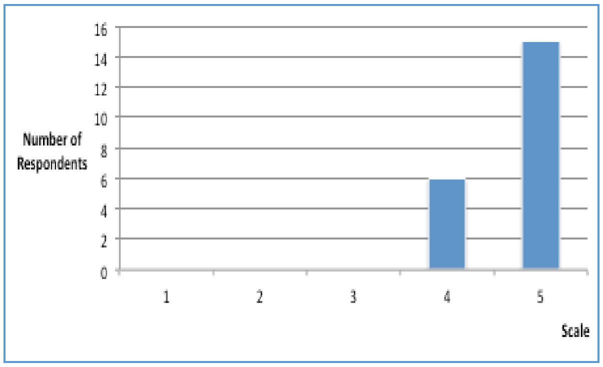

Figure 12. Evaluation of Managing Feedback.

7.2 Workshop

The workshop aims to evaluate the enhancements that we proposed to improve the current CRE platforms. A group of four persons interested in the process of CRE platforms are met together.

In general, the participants believe that the proposed enhancements can improve the current workflow. When asking about their opinions in each of the five enhancements, we received remarks about enhancements A, C and D.

All the respondents think it is a good idea to take some influential factors into consideration such as the crowd's competence as illustrated in enhancements A (Developing Crowd's Profile). However, they suggest using other factors such as the time zone as it could help utilize and manage diversity and improve auto-developing project schedules.

With regard to enhancement C (Creating Project Plan), some suggest applying mechanisms to easily help crowd manager to estimate the task duration and the optimal number of team members needed to complete a task. This can involve using heuristic data of crowd's efforts in similar tasks.

Furthermore, with regard to enhancement D (Managing Requirements Overlapping and Duplicates), they suggest adding a layer between the crowd and the crowd manager (i.e. crowd team leader) in order to eliminate the crowd manager's workload.

8. DISCUSSION AND CONCLUSIONS

Introducing crowdsourcing into requirements engineering provides valuable benefits such as leading to more relevant and creative requirements, increasing the ability to understand the foundation for requirements and having complete solutions.

This work contributes in narrowing the research gap regarding studying CRE process. A study was performed to assess the current crowdsourced requirements engineering platforms. Based on this assessment, a set of limitations have been identified. These limitations confirm the general observations described in the literature about the weaknesses of the current CRE process [2] [10] [3]. Furthermore, the evaluation of the proposed enhancements to overcome the limitations shows that they are sound and practical.

It can be concluded from both the questionnaire and workshop that our proposed solution has some advantages in supporting the current CRE platforms' workflow. It shows that current CRE platforms still have space for improvements.

Yet, our analysis provides only limited processes on coordination, management, scheduling and planning. More analysis is required to extend the work and simplify some processes such as assigning task and developing project schedule.

In addition, this study uses small sample in evaluation through the questionnaire and workshop. Although the findings seem interesting, more work on evaluation is required. This includes integrating the proposed enhancements in CRE platforms and using them in real projects. Despite the mentioned limitations, we believe the proposed enhancements provide useful contribution that can be a good start for further research in the area.

For the future work of our research, we suggest having more improvements in crowdsourcing for requirements engineering that can extend our proposed enhancements. For example, ensuring crowd manager's availability and productivity, considering some other factors in categorizations such as requirements dependency, and simplifying communication, and coordination between crowd and crowd manager.

REFERENCES:

[1] P. K. Srivastava and R. Sharma, "Crowdsourcing to elicit requirements for MyERP application," in *1st International Workshop on Crowd-Based Requirements Engineering, CrowdRE 2015 - Proceedings*, 2015.

[2] A. Adepetu, K. Ahmed, and Y. Al Abd, "CrowdREquire: A Requirements Engineering Crowdsourcing Platform," *2012 AAAI Spring …*, 2012.

[3] R. Snijders, F. Dalpiaz, M. Hosseini, A. Shahri, and R. Ali, "Crowd-centric requirements engineering," in *Proceedings - 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, UCC 2014*, 2014.

[4] I. Sommerville and P. Sawyer, *Requirements Engineering: A Good Practice Guide*. 1997.

[5] B. Nuseibeh and S. Easterbrook, "Requirements engineering: a roadmap," *Proc. Conf. Futur. Softw. Eng. - ICSE '00*, 2000.

[6] W. Maalej and A. K. Thurimella, "An Introduction to Requirements Knowledge," in *Managing Requirements Knowledge*, 2013.

[7] T. S. Sindlinger, "Crowdsourcing: Why the Power of the Crowd is Driving the Future of Business," *American Journal of Health-System Pharmacy*. 2010.

[8] J. Howe, "The Rise of Crowdsourcing," *Wired Mag.*, 2006.

[9] M. Vuković, "Crowdsourcing for enterprises," in *SERVICES 2009 - 5th 2009 World Congress on Services*, 2009.

[10] M. Hosseini, A. Shahri, K. Phalp, J. Taylor, R. Ali, and F. Dalpiaz, "Configuring crowdsourcing for requirements elicitation," in *Proceedings - International Conference on Research Challenges in Information Science*,

2015.

[11] S. L. Lim, D. Damian, and A. Finkelstein, "StakeSource2.0: using social networks of stakeholders to identify and prioritise requirements," *2011 33rd Int. Conf. Softw. Eng.*, 2011.

[12] S. L. Lim and A. Finkelstein, "StakeRare: Using social networks and collaborative filtering for large-scale requirements elicitation," *IEEE Trans. Softw. Eng.*, 2012.

[13] 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.

[14] H. Wang, Y. Wang, and J. Wang, "A participant recruitment framework for crowdsourcing based software requirement acquisition," in *Proceedings - 2014 IEEE 9th International Conference on Global Software Engineering, ICGSE 2014*, 2014.

[15] T. D. Breaux and F. Schaub, "Scaling requirements extraction to the crowd: Experiments with privacy policies," in *2014 IEEE 22nd International Requirements Engineering Conference, RE 2014 - Proceedings*, 2014.

[16] N. M. Ochara, D. Asmelash, and S. Mlay, "Group Think Decision Making Deficiency in the Requirements Engineering Process: Towards a Crowdsourcing Model," *Innov. Vis. 2020 Sustain. Growth, Entrep. Econ. Dev. Vols 1-4*, 2012.

[17] M. Levy, I. Hadar, and D. Te'eni, "A gradual approach to crowd-based requirements engineering: The case of conference online social networks," in *1st International Workshop on Crowd-Based Requirements Engineering, CrowdRE 2015 - Proceedings*, 2015.

[18] N. Seyff and F. Graf, "User-Driven Requirements Engineering for Mobile Social Software.," in *Software Engineering (Workshops)*, 2010, pp. 503–512.

[19] E. C. Groen, J. Doerr, and S. Adam, "Towards crowd-based requirements engineering a research preview," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015.

[20] W. Maalej, M. Nayebi, T. Johann, and G. Ruhe, "Toward data-driven requirements engineering," *IEEE Softw.*, vol. 33, no. 1, pp. 48–54, 2016.

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

[22] S. L. Lim, D. Quercia, and A. Finkelstein, "StakeSource: harnessing the power of crowdsourcing and social networks in stakeholder analysis," *2010 ACM/IEEE 32nd Int. Conf. Softw. Eng.*, 2010.

[23] N. Kukreja and B. Boehm, "Process implications of social networking-based requirements negotiation tools," in *2012 International Conference on Software and System Process, ICSSP 2012 - Proceedings*, 2012.

[24] J. Fernandes, D. Duarte, C. Ribeiro, C. Farinha, J. M. Pereira, and M. M. Da Silva, "IThink : A game-based approach towards improving collaboration and participation in requirement elicitation," in *Procedia Computer Science*, 2012.

[25] D. Renzel, M. Behrendt, R. Klamma, and M. Jarke, "Requirements Bazaar: Social requirements engineering for community-driven innovation," in *2013 21st IEEE International Requirements Engineering Conference, RE 2013 - Proceedings*, 2013.

[26] R. Snijders, F. Dalpiaz, S. Brinkkemper, M. Hosseini, R. Ali, and A. Özüm, "REfine: A gamified platform for participatory requirements engineering," in *1st International Workshop on Crowd-Based Requirements Engineering, CrowdRE 2015 - Proceedings*, 2015.