

# A Collaborative Requirements Elicitation Model For Crowdsourcing Platforms

Chantal Mukundwa\*, Seok-Won Lee\*

## Abstract

Crowdsourcing is growing its interests in designing where various designers work independently to a given design task. Recent researchers discovered that collaboration by sharing designs among designers helps to produce high-quality designs. However, design task may still be hard even with that collaboration in case the requirements are not well-defined. Most customers sometimes do not know what they really want and do not know how to clearly define the requirements. Consequently, the lack of requirements creates issues on designers, such as spending much time and effort on collecting requirements alone or from the customers. The designers even end up missing important necessities to complete their tasks. To address this issue, we proposed a collaborative requirements elicitation method that supports designers who are working on the same task. We developed CREFD (Collaborative Requirements Elicitation For Designers and Developers) tool to enable designers collaboratively provide requirements, identify dependencies, add annotations and votes to the provided requirements. We performed the hypothetical and empirical evaluations to test and compare the proposed method with one of the existing elicitation methods, the results show that the proposed method helps in collecting accepted and well-organized requirements better than individual requirements elicitation.

▶ Keyword : Crowdsourcing, requirements elicitation, collaboration, design process

## I. Introduction

Recently, there has been an ongoing interest in crowdsourcing where both the individuals and companies/organizations benefit various skills and creativities from the crowd. The term crowdsourcing has been coined by Jeff Howe in 2006 [1], and since then this idea has been successfully applied to various areas [18] in different forms such as crowd contests (eg: designcrowd), crowdfunding (eg: Kickstarter), microtasks (eg: Amazon mechanical turk) etc.

Some of the crowd contest platforms are platforms used for design process/task which enable a customer to post a task to the platform and designers from different locations

create independent designs to compete each other, then only a single designer whose solution gets selected by the customer as the best gets the compensation.

Within those platforms for the tasks, designers may easily make unacceptable designs or fail to understand what to do based on requirements provided by the customers because sometimes they are not clearly defined. In [6–9] some examples show how customers posted the design task on the platform and how they communicate with designers. The examples show that when a customer posts a task, sometimes the task does not have well defined and organized requirements which causes:

- 
- First Author: Chantal Mukundwa, Corresponding Author: Seok-Won Lee
  - \*Chantal Mukundwa (chantygi@ajou.ac.kr), Dept. of Computer Engineering, Ajou University
  - \*Seok-Won Lee (leesw@ajou.ac.kr), Dept. of Computer Engineering, Ajou University
  - Received: 2018. 12. 11, Revised: 2019. 03. 11, Accepted: 2019. 03. 11.
  - This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2017R1D1A1B03034279).

- Designers or participants may spend more time and effort, unnecessarily, to complete a design.
- Designers sometime end up missing mandatory requirements to complete their tasks efficiently. In fact, they make unacceptable designs.
- Designers ask too many questions to the customer in order to exactly know what to do; however, sometimes the customer does not provide the answers to the asked questions.

For such issues, this paper aims at designing, evaluating, and analyzing a cooperative requirements elicitation method that can support designers. This will allow them to collaborate with no need for them to meet physically or synchronize their time zone in order to collect in due time accepted requirements that are needed to their designing task. Additionally, customers also benefit by the proposed method which helps them to easily communicated with the designers.

The proposed method can be implemented as an add-on to current platforms to enable both designers and customers to understand requirements that are important and needed to a given task and organize those identified requirements before starting the design task. The target is to make a method that helps the customer gets a design that has all the important features.

## II. Related Work

Various studies have discovered the role of requirements elicitation as the better way to meet the customer's needs, and others have identified that collaboration should be considered to effectively complete the design with high quality [17].

As requirements are the expectations of the stakeholders or customers which specify how the software should behave [21], thus, requirements elicitation phase is an important step and should be considered to support other phases in order to solve different issues that may be faced when requirements are not well defined. According to [13], the researcher has identified that a project can easily fail when the requirements are badly defined. And the works [5] [14] identified the reason of projects failure as the lack of the customers knowledge to better specify what they really want. Due to the lack of well-defined requirements,

designers come up with the outcome which does not meet the customer expectations or needs.

In the past years, traditional requirements elicitation methods have been used to elicit the requirements; however, with these methods, it is hard to involve a large number of users [3]. In case of a big project which needs a lot of stakeholders, it may be hard to reach all of them with traditional methods.

Also, text and data mining techniques are used to facilitate requirements capturing and classifying processes [11, 12]; however, it is still hard with these techniques to handle peculiarities of natural languages. The idea of collecting requirements using crowdsourcing concept has been discussed within different researches to benefit the knowledge of the crowd and handle those peculiarities or unusual texts (poor writing, poor spelling, lack of necessary punctuations, mixing multiple languages etc.) which may be caused by users.

In [4], researchers design a model that helps identification of requirements through feedbacks which are annotated by the crowd. With this model, crowd annotate feedback using predefined categories and create a new category. However, in order to avoid duplicate entries, every new entry needs to be confirmed by requirements engineers. This model cannot be used to gather the new requirements, but instead, it is used to organize the requirements in their respective categories.

The work in [2] has proposed a conceptual requirements engineering crowdsourcing platform where the crowd help customers (individual and companies) in finding the best requirements specification for their proposed tasks and projects; however, this platform is useful when the task is well described. Otherwise; it is hard for the crowd to contribute or to understand well what to do. In [10], the authors proposed an approach that uses crowd's knowledge to elicit requirements for dynamically adaptive systems using questionnaire technique. This approach relies on both requirements engineers, known crowd (experts) and the unknown crowd (potential end users) which may limit potential end users to contribute alone with this approach.

Also, although this research included the requirement elicitation in the steps, it is still not easy to come up with a questionnaire that has enough question to come up with all needed requirements in due time.

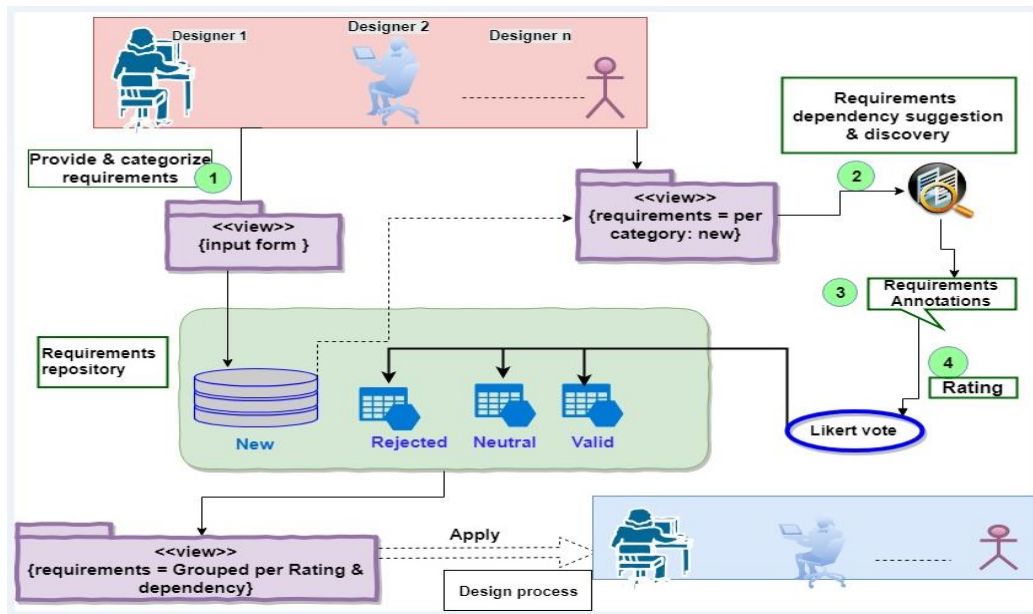


Fig. 1. System Overview

### III. Proposed Methodology

This paper proposes a method that supports designers to collect, share, annotate, and validate requirements. The figure 1 illustrates an overview of the proposed method.

#### 1. The proposed methodology users (workers)

The proposed methodology relies on users to provide the initial, non-rated, requirements and also run the dependency management as well as voting on requirements. Based on the type of task, the users can be from different backgrounds based on their area of expertise. In this study, we concentrated on software designers, but the roles can expand to more than just designers.

#### 2. The proposed methodology components

##### 2.1 Requirements repository

The requirements repository is organized by requirements ratings earned per each as of:

- New requirements: Default for every new provided requirement, not voted on yet.
- Voted requirements: Rejected, neutral and accepted.

##### 2.2 Views

The Views are interfaces generated by the repository controller (It is an MVC design) and plays two roles: collect the input from the users and display the content to the users.

- Form view: Collect input from users.

This view is generated by the repository controller during the time to provide new requirements is a form that allows the system to collect new requirements (with their types) and the collected content is stored in the repository.

- Displayer view: Display the content to the users.

This view is also generated from the repository controller and changes based on the active time frame (requirements provision, or voting, or when elicitation

phase is completed) and the content in the requirement repository. The first displayer view is requirements per category view which displays all the requirements categorized as “new” to the user. This view then allows the user to vote, annotate, and identify the dependency as helped by the system. It is activated and accessible during the voting time only. The second displayer view is requirements per rating and dependency view which presents to the user the results of the requirement elicitation process; thus, the user can proceed with the designing and developments tasks. It is available and accessible only after the completion of the process of rating (voting) the requirements.

#### 3. The proposed methodology steps

There are four steps in the proposed methodology as shown in table 1, which summarizes the step-activities of the proposed requirements elicitation methodology.

Table 1. Proposed method activities

Steps	Activities
1	Users provide requirements by their types
2	Managing dependencies
3	Users provide annotations/comments (if necessary)
4	Users rate their own and peers' requirements

**Step 1: Users provide requirements by types.**

Within this method there will be two ways to provide requirements. First way, each participant provides requirements which are viewed by others as well. Participant provides either functional, nonfunctional, business, or user requirements. This helps to have well-organized requirements. Second way, the participant annotates others' requirements.

While providing requirements in the first way, participants need to add requirement title, requirement description, and categorize that requirement by choosing from available requirements types as shown in the figure 2.

Participants can refer to the available explanations (in helper) about different types of requirements (functional, nonfunctional, business and user) to better categorize the requirements.

Fig. 2. Providing requirements form

**Step 2: Managing dependencies**

Capturing dependencies between requirements is very important to facilitate the understanding of requirements relationships. Some of the dependencies types are: refinement, constraints, precondition, satisfaction, similarity, and etc. [15]. In fact, dependencies are likely to occur among requirements when different people are collaborating [16].

In this step, the participants are presented with all provided requirements and the view enables them to relate each requirement to its dependent. In our case study, only one type of requirements dependencies (similarity) was implemented and tested.

The similar requirements are identified in our proposed method by participant users where each participant before submitting the vote on a given requirement. They are asked to identify similar requirements by comparing the selected requirement with the requirements that they have already voted in the same pool. After, all users have provided their identified dependencies. The system provides a view for the presentation of the identified similarities from all participants using the displayer view. The table 2 shows a snippet example of how the similar requirements are arranged.

Table 2. Similar-dependency presentation example

Requirement title	Requirement description	Duplicated? <b>Condition:</b> If yes Show a list of similar requirements
Usability	System should be easy to use	Yes <b>Selected requirements:</b> ● Title: Usability ● Description: System should be easy to use.
		<b>Similar to:</b> ● Title: Usability ● Description: All operations should be learnable in short time to facilitate the system users.

**Step 3: Users provide annotations/ comments**

Annotation is a note of explanation or comment that is added to a text or diagram to make it more understandable or explainable. Annotations can support or challenge the requirement provided. In our proposed method, participants are allowed to annotate their own requirements or others' requirements. This may help participants making new requirements based on provided annotations, or make clearer a requirement provided by others, and help them to the make right choice during the voting process.

**Step 4: Users rate their own and peers' requirements**

The Likert scale uses fixed choice response formats and is designed to measure opinions. Our methodology uses agreement Likert scale format (Strongly Agree=5, Agree=4, Undecided=3, Disagree=2, Strongly Disagree=1) for validating the requirements provided by different participants. After the voting process, all requirements are categorized based participant's role and grouped into three categories which are:

- Accepted: If the requirement averaged score of "agree" or "strongly agree"
- Neutral: If the requirement averaged a score of "Neither agree nor disagree"

- Rejected: If requirement averaged score of “disagree” or “strongly disagree”

## IV. Evaluation

We used two evaluations in our work. First, the proposed methodology was evaluated hypothetically with the help of study questions and propositions. Then the propositions were supported by gathered evidences. Furthermore, an empirical study was performed to compare our methodology with one of existing method.

### 1. Testing tool: CREFD

We developed a Collaborative Requirements Elicitation For Designers and developers (CREFD) tool for evaluation of our proposed method where various participants collaborate in requirements elicitation for a given design task. The tool is developed as a web application which helps the users to easily access the application from any computer.

The tool is used in two-time frames as according to the proposed methodology:

- First, there is a limited time to provide requirements; however, participants are not limited to provide requirements until the given time for providing is over.
- The second, participants start the voting process where they are allowed to annotate provided requirements and identify duplicate requirements.

### 2. Hypothetical evaluation

The hypothetical evaluation helps to prove the feasibility of the proposed method by checking if it answers the study questions. We adopted the validation methodology recommended in [19].

#### 2.1 Study questions and propositions

General proposition of the proposed methodology is “the proposed method can achieve its goals because it enables various users collaborate in eliciting and validating requirements”. Below are the 3 study questions and 7 specific propositions (SP).

Q1. How can the proposed method help designers collect many requirements without spending much effort unnecessarily?

- SP1.1 Each participant is allowed to provide new

requirements and annotate requirements provided by others.

- SP1.2 The number of accepted requirements provided by a group of people outnumber the requirements given by an individual person.

Q2. How can the proposed method help designers collect accepted (accurate) requirements?

- SP2.1 The proposed method helps to identify similar requirements.
- SP2.2 Participants can vote requirements to support identification of accepted requirements.

Q3. How and why the proposed method helps people from different places collaborate to generate organized requirements without having to meet in-person nor have to synchronize their timezones?

- SP3.1 Participants can easily categorize the requirements by referring to the explanations of different types of requirements available in the proposed method tool.
- SP3.2 With the help of developed tool, we can identify each participant’s contribution.
- SP3.3 The proposed tool gives timeframe for each activity (providing requirements and voting process). For each, the time frame is set by the administrator which is done after observing the availability of each participant in order to pick a timeframe enough for every participant to give their requirements.

#### 2.2 Units of Analysis

In our study, we have seven units of analysis which used as the evidences to support the propositions. With our proposed solution, we are able to collect:

- Annotated requirements
- Categorized requirements
- Effort spent (contribution by each participant)
- Similar requirements
- Requirements types
- Total accepted requirements against total individual requirements.
- Percentage of user’s participation in both providing requirements and voting on requirements.

#### 2.3 Linking Data

To link the generated unit of analysis and study propositions, connecting both objects is important. Table 3 shows how the data are linked to proposition.

Table 3. Linking Data to Proposition

Study Propositions Descriptions	Evidence
SP1.1 Each participant is allowed to provide new requirements and annotating requirements provided by others.	Annotated requirements
SP1.2 The number of requirements provided by a group of people outnumber the requirements given by an individual person.	Total accepted requirements against total individual requirements
SP2.1 The proposed method helps to identify similar requirements.	Similar requirements
SP2.2 Participants can vote requirements to support identification of accepted requirements which are based on voting score.	Categorized requirements
SP3.1 Participants can easily categorize the requirements by referring to the explanations of different types of requirements available in the proposed method tool.	Requirements types
SP3.2 With the help of developed tool, we can identify each participant's contribution.	Effort spent(contribution by each participant)
SP3.3 The proposed tool gives timeframe for each activity (providing requirements and voting process). For each, the time frame is set by the administrator which is done after observing the availability of each participant in order to pick a timeframe enough for every participant to give their requirements.	Percentage of users participation in both providing requirements and voting on requirements

**2.4 Summary of Result**

To gather the evidences, we match the experiment result of each step of the proposed methodology to the propositions. Our case study task was to collect all relevant requirements for building a website for a primary school. We applied our methodology to this task where 17 participants (students from computer engineering) have participated. The table 4 below shows the evidence captured with supporting propositions.

**3. Empirical evaluation**

**3.1 Purpose of the Study**

We conducted the empirical study to evaluate the applicability, usability, and scalability of the proposed methodology. This work is related to elicitation of requirements from users through collaboration where users act as designers and developers who collect requirements that are needed to build their design tasks. We also aimed to evaluate the requirements collected with other traditional method and our proposed method and then make comparison.

**3.2 Measures**

As the purpose of this work is to evaluate the applicability, usability and scalability of our proposed

methodology, we examined the number of requirements elicited (with existing requirements elicitation techniques / with the proposed methodology) to evaluate the result of this empirical study.

**3.3 Method**

We introduced the objective of our study to participants before starting their participations. The participants contributed with both the survey and proposed method where they were given the same task on both methods. Finally we documented the requirements elicited with all methods.

**3.4 Task**

The task was explained to the participants in order to make them understand what to do. However, only few requirements were listed in the task. The task was given to 6 people who have background in web designing and developing. Those people were asked to work independently and provide all necessary requirements that should be considered while building "Home alone safety for children" application.

After collecting those requirements, the same task was given to 4 people who participated in requirement elicitation using survey, and, 2 more new people were added.

We asked all the 6 to work using CREFD tool and then evaluated the impacts and differences of requirements provided in both methods.

**3.5 Results and analysis**

Figure 3 and 4 show the summary of the requirements elicited by survey (independent work) and the requirements elicited using the proposed CREFD tool (collaborative work).

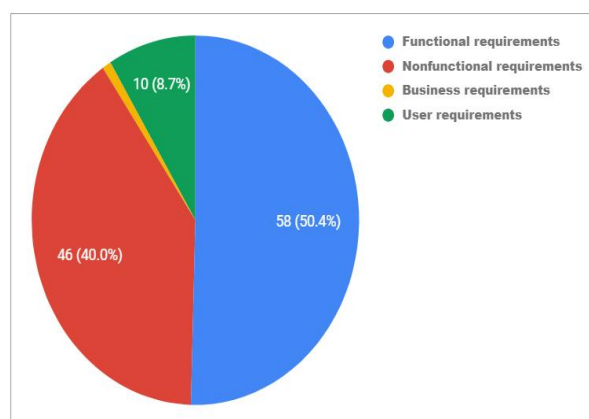


Fig. 3. Requirements elicited with the independent work

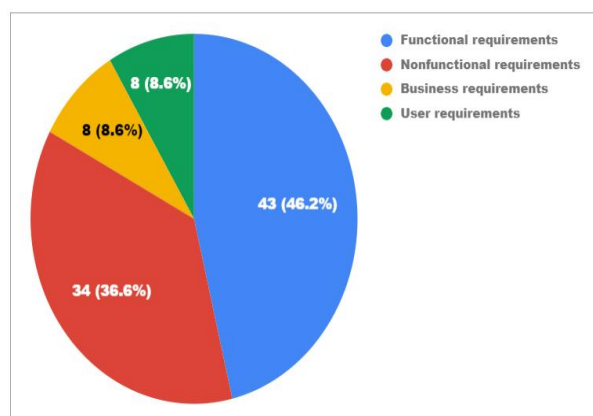


Fig. 4. Requirements elicited with the proposed methodology work

The figure 3 shows a combination of all requirements gathered with 6 users with independent work.

With the independent work, some participants struggled to find out enough requirements, there are even users who elicited only 6 and 11 requirements. The best user who elicited many requirements with independent work, his amount of requirements is still lesser than the requirements with the proposed method. Regarding the duplicates which are very common within the collaboration, independent work has more

duplicates (34 duplicates requirements) compare to the proposed method which has only 7 duplicates requirements. With the proposed method, there is a chance to avoid duplications since the participants can see the requirements provided by others.

With the proposed methodology, participants categorized the requirements by types easily after learning about the requirements types from the proposed tool, and they rated the requirements by their level of importance. From the total contributions of the proposed method, 76 requirements were accepted, 14 neutral and 3 were rejected, but, with the independent work, there was no way for participants to rate the requirements together.

After collecting requirements with both methods, we asked the participants feedback to our methodology regarding advantages, weaknesses, and to compare it with the existing method.

#### Asked questions:

- Mention the advantages realized while using the collaborative tool (our proposed methodology).
- Mention the weakness detected while using the collaborative tool.

Table 4. Evidence collection

Activities	Captured Evidence	Supported Propositions
Step 1: Users provide requirements by their types	CREFD allowed to collect a variety of requirements. Requirements were elicited in types User requirements: 12.09% Business requirements: 9.89% Functional requirements: 39.56% Non-Functional requirements: 38.46%	SP1.1, SP3.1
Step 2: Managing dependencies	Similar requirements are identified by considering, requirements title or requirements description. For example:  Requirement title: Security ● Requirement description from user x This is more crucial since there will be the accessibility of private information, thus users need to be authorized. ● Requirement description from user y Both parents and students should be registered in order to access students' records.  Requirement title: Usability Requirement description from user m System should be easy to use. Requirement description from user n All operations should be learnable in short time to facilitate the system users.	SP2.1, SP3.2
Step 3: Users provide annotations/ comments (if necessary)	With our tool, users can provide annotations that support or reject the idea. For example: Requirements title: Maintainability Provided annotation: ● Website should notify registered users about emergency case or in case of maintenance	SP 1.1
Step 4: Users rate their own and peers' requirements	After voting, requirements are arranged as accepted, neutral, rejected. ● Elicited accepted: 69, Elicited neutral: 11 ● Elicited rejected: 0, Not voted requirements: 11 And votes are displayed based on user's role. ● Developers votes Designers votes ● Average of both developers and designers	SP1.2, SP2.2 SP3.2, SP3.3

**Advantages:**

The participants mentioned that the proposed tool helps to discover various ideas in an easy and quick way. It enables to benefits ideas from highly skilled people, helps to get organized ideas. They also found that the tool is easy to use and enhance the innovation.

**Weaknesses:**

Duplicates take time to discover. Some participants had difficult to be specific on which requirements should be used due to many ideas. They claimed that automated duplicates discovery would have been more helpful.

## V. Discussion

In this section, we analyze the quality attribute of our proposed methodology.

### 1. Application

We showed how people can work together to provide and validate the requirements needed for designing tasks such as web design using the proposed method; however, this method can be applicable for gathering requirements for developing different systems and applications. For example, the proposed methodology can be used in a company where a team wants to collect requirements that may be applicable to a certain task. Also, an individual may use it to gather the requirements.

### 2. Scalability

The proposed methodology can be extended to different projects where people would have points to discuss before starting the project. We focused on web design task because this task takes time to be completed, and most of the time this task needs more than single user to be done properly. [20] shows that a project may easily gets fail when the requirements are not well defined and arranged. To avoid this issue, various users need to discuss on needed requirements before starting implementing their solution. In future work direction, we plan to explore more features that may support different types of designs and non-designs projects.

### 3. Usability

CREFD is user-friendly for the users. The interfaces are simple, and the users are guided on how to categorize requirements by reading different types of requirements.

Although participants wish the automated support to reduce their efforts in the dependencies managing, participants mentioned that the proposed methodology is very useful to elicit and validate requirements.

## VI. Conclusions and Future Work

We have emphasized the importance of collaboration in requirements elicitation process, which contribute a lot to making a good design by discovering relevant requirements in due time. With this methodology, designers and developers can provide and access requirements provided by others, vote and annotate both their own and other's requirements. The purpose of annotations was to support or reject the proposed requirement. Our objective was to help also the customers to better understand their idea in different angles and provide comments where necessary, which guides each participant to make a decision while selecting requirements to be used to their designs.

The proposed methodology can be simple for both the crowdsourcing platform to implement and for the participant to use; although, as of now, crowdsourcing platforms are yet to implement such collaborative requirements elicitation methodology that will help both crowdsource platforms and the crowds within the design process.

Although our case study was on design tasks, it can be extended to further tasks. We have developed a tool, CREFD, that helps to evaluate the feasibility and efficiency of the proposed method. With the CREFD tool, the designers and developers were able to collaborate in gathering and validating requirements where each participant votes requirements in order to rate them by their level of importance according to the rating of 1 to 5. The least important requirement is rated as 1 and the most important is rated as 5. We analyzed the impact of CREFD tool by comparing it with independent work. We found that the participants preferred working with CREFD tool than independent work.

We intend to extend our methodology by combining it with natural language processing (NLP) techniques to support requirements analysis and categorization where a system will suggest the type of requirements for users based on keywords. Our propose methodology presents some limitations which should be covered in the future. In the future, we plan to discover the requirements dependencies by both humans and



automatic techniques, and the requirements will be ranked differently according to the user's profile. Means that the users' capabilities will be identified before working on the requirements elicitation process.

## REFERENCES

- [1] Lynch, Alec. "Crowdsourcing is not new—The History of Crowdsourcing (1714 to 2010)." DesignCrowd. N.p., 2010. Web. 11 Apr. 2018.
- [2] Adepetu, Adedamola, et al. "CrowdREquire: A Requirements Engineering Crowdsourcing Platform." AAAI Spring Symposium: Wisdom of the Crowd. 2012.
- [3] Groen, Eduard C. et al. "The crowd in requirements engineering: The landscape and challenges." *IEEE software* 34.2 (2017): 44–52.
- [4] Hosseini, Mahmood, et al. "CRAFT: A Crowd-Annotated Feedback Technique." 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW). IEEE, 2017.
- [5] Mierke, David. "3 Reasons You Can'T Just Ask Customers What They Want". Techcrunch, 2018, <https://techcrunch.com/2014/11/15/3-reasons-why-you-cant-just-ask-customers-what-they-want/>. Accessed 2 May 2018.
- [6] Jobs, Design et al. "Stress Coach Website On Designcrowd.Com". Designcrowd, 2018, <https://jobs.designcrowd.com/job.aspx?id=3485184>. Accessed 9 July 2018.
- [7] Jobs, Design et al. "Web Design Job – Condo Debts Collection Business Needs A Web Design On Designcrowd.Com". Designcrowd, 2018, <https://jobs.designcrowd.com/job.aspx?id=3488383>. Accessed 4 Aug 2018.
- [8] Jobs, Design et al. "Website For LORD On Designcrowd.Com". <https://jobs.designcrowd.com/job.aspx?id=1790423>. Accessed 4 June 2018.
- [9] Jobs, Design et al. "New Fresh Website Required To Go With Exciting Expansion And Re-Brand Vehicle Repair Garage In UK On Designcrowd.Com". <https://jobs.designcrowd.com/job.aspx?id=3032829>. Accessed 18 July 2018.
- [10] Munante, Denisse, et al. "Gathering Requirements for Software Configuration from the Crowd." 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW). IEEE, 2017.
- [11] Castro-Herrera, Carlos, et al. "Using data mining and recommender systems to facilitate large-scale, open, and inclusive requirements elicitation processes." *International Requirements Engineering*, 2008. RE'08. 16th IEEE. IEEE, 2008.
- [12] Palmeira, Clay, et al. "A requirements elicitation and analysis aided by text mining." *International Journal of Computer Science and Network Security (IJCSNS)* 12.6 (2012): 122.
- [13] "Why Software Fails". *IEEE Spectrum: Technology, Engineering, And Science News*, 2018, <https://spectrum.ieee.org/computing/software/why-software-fails>. Accessed 2 Aug 2018.
- [14] Sangeeta, and Seok Won Lee. "REASSURE: Requirements elicitation for adaptive socio technical systems using repertory grid." *Information and Software Technology* 87 (2017): 160–179.
- [15] Dahlstedt, G. "Requirements Interdependencies—a research framework." no. July (2001).
- [16] Ambler, Scott, and View & rarr: "Managing Requirements Dependencies Between Agile Teams." *The Disciplined Agile (DA) Framework*. N.p., 2014. Web. 21 Aug. 2018.
- [17] Park, Cheong Ha, et al. "Crowd vs. crowd: large-scale cooperative design through open team competition." *Proceedings of the 2013 conference on Computer supported cooperative work*. ACM, 2013.
- [18] Alan, David. "Crowdsourcing For Dummies." O'Reilly | Safari. N.p., 2018. Web. 4 May 2018.
- [19] Lee, Seok Won, and David C. Rine. "Case Study Methodology Designed Research in Software Engineering Methodology Validation." *SEKE*. 2004.
- [20] "Why Software Fails". *IEEE Spectrum: Technology, Engineering, And Science News*, 2018, <https://spectrum.ieee.org/computing/software/why-software-fails>. Accessed 2 Aug 2018.
- [21] Iqbal, Tabbassum, and Mohammad Suaib. "Requirement elicitation technique:—a review paper." *Int. J. Comput. Math. Sci* 3.9 (2014).

## Authors



Chantal Mukundwa received BS in Engineering from University of Rwanda College of Science and Technology, Department of Computer Engineering and Information Technology, Rwanda in 2014 and MS in Computer Science and

Engineering from Ajou University, South Korea in 2019.



Seok-Won Lee received the BS in Computer Science from Dongkuk University at Seoul, Republic of Korea, MS in Computer Science from University of Pittsburgh at Pittsburgh, Pennsylvania, and PhD in Information Technology from George Mason University

at Fairfax, Virginia. He is currently a Full Professor in the Department of Software and Computer Engineering at Ajou University, Republic of Korea since 2012. His areas of specialization include software engineering with specific expertise in ontological requirements engineering and domain modeling, and knowledge engineering with specific expertise in knowledge acquisition, machine learning and knowledge-based systems.