Universiteit
Leiden
The Netherlands

Involving the crowd in Requirements Engineering

Louise Kubatz

Supervisors:

T.D. Offerman MSc & P. van Leeuwen MBA & Dr. A.W. Laarman

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

11/08/2021

Abstract

Crowd-based Requirements Engineering is a new type of Requirements Engineering. The main activities of Requirements Engineering are the same, but the people used in the process of gaining requirements are different. In crowd-based Requirements Engineering developers use a crowd which is a large group of current or potential users. The crowd is continuously involved in the process and after the implementation of the product. The crowd provides user feedback. Crowd-based Requirements Engineering also uses crowd-based monitoring of software products to obtain new requirements. There is not much known about the successful execution. Therefore it is difficult to state the precise effects. To know more about the current status of crowd-based Requirements Engineering in practice, we answer the following research question: 'What are the lessons learned from using crowd-based Requirements Engineering?'. To answer this question we conducted an exploratory, qualitative research. We performed 12 interviews with participants that are involved in the Requirements Engineering process. We used thematic analysis to analyze the transcribed interviews. This resulted in a list of codified terms and their frequencies. Out of these results, the lessons learned are interpreted. The lessons learned of crowd-based Requirements Engineering are that for these participants there isn't any awareness for the concept of crowd-based Requirements Engineering in practice. The participants mostly use traditional Requirements Engineering techniques in their process. Although, we do see an aspect of crowd-based Requirements Engineering in the user involvement. We see that the stakeholders or users unconsciously continuously are involved by the participants. It is mentioned by participants that more involvement of stakeholders or users can result in a better product and better understanding users' feedback. The lesson learned in this is that there might be expectations of these stakeholders or users which need to be managed. It would be interesting for further research to find out what techniques are used by organizations to manage these expectations.

Contents

1 Introduction

The world is currently in the Fourth Industrial Revolution. Software is getting more intelligent each day. Companies need to adjust to these innovations in order to keep up with their competition in the market. It is important to have software that satisfies the customer in order to create competitive advantages and gain business growth. To create software that satisfies the customers needs, there is a process prior to the software development. This process is called Requirements Engineering. Requirements Engineering results in effective product development by preventing errors in an early stage, getting a clear understanding about the product, gaining higher product quality and better customer satisfaction. There is a new concept of Requirements Engineering, called crowd-based Requirements Engineering.

Crowd-based Requirements Engineering is an addition to the traditional Requirements Engineering. The main activities of Requirements Engineering are the same, but the people used in the process of gathering requirements are different. In crowd-based Requirements Engineering developers use a crowd which is a large group of current or potential users[GSA+17]. The crowd is continuously involved in the process and after the implementation of the product. The crowd provides user feedback.

1.1 Problem Statement

The concept of crowd-based Requirements Engineering seems promising. Although there are some complications. First of all, the quality of requirements cannot be ensured. It is unknown what conditions and quality measurements are needed for the crowd in order to deliver relevant requirements [KLWA19]. Second, the privacy of crowd-members can be a restriction for the implementation of crowd-based Requirements Engineering. Furthermore, motivating crowd-members can result in affecting the user's feedback. Also analyzing the feedback comes with difficulties. Sometimes the users are anonymous, so it is hard to identify the different subgroups[GDA15]. Identifying all the relevant data and finding the feedback that occurs multiple times are hard. Additionally, a minority of users with good feedback must not be overlooked[GSA+17]. All these complications raise questions about the effectiveness of crowd-based Requirements Engineering. It is stated that there is not much known about the successfully execution of crowd-based Requirements Engineering[GSA+17]. Therefore, it is difficult to state the precise effects of crowd-based Requirements Engineering.

In this research we want to focus on this last problem. The problem is that we don't know the effects of crowd-based Requirements Engineering, because we don't know much about the successful execution of crowd-based Requirements Engineering. To know more about the possible effects of crowd-based Requirements Engineering we need to learn more about the different Requirements Engineering practices and get a better view of where we are in practice with crowd-based Requirements Engineering. This research is done with organizations that use Requirements Engineering to build software or an IT-related product. There is no further distinction made between organizations. Along the way we ask what the participants of this research have experienced when using crowd-based Requirements Engineering and therefore figure out what the lessons they have learned. This will give us more insights in the concept of crowd-based Requirements Engineering. We won't be able to state the precise effects of crowd-based Requirements Engineering. This is

simply because we don't know what the current status of crowd-based Requirements Engineering in practice is. Therefore, this research will be focused on what the lessons learned are from applying crowd-based Requirements Engineering. The research question is 'What are the lessons learned of using crowd-based Requirements Engineering?'.

This question is relevant because crowd-based Requirements Engineering could lead to benefits for the users of software product. This can result in business growth and competitive advantages for organizations. The question is also relevant because it should be easier nowadays to use crowd-based Requirements Engineering. Automated techniques and machine intelligence are necessary to derive requirements from the large set of data[MAS17]. These algorithms and software are getting more intelligent each day. This should make the process of crowd-based Requirements Engineering more effective. Crowd-based Requirements Engineering is new and experimental. Therefore it is necessary to get more insights in this concept to reveal its potential.

The following sub-questions are asked in order to help answer the research question.

- How is Requirements Engineering implemented in practice?
- How has the process of Requirements Engineering transformed over time?
- What are the perceived effects of using more crowd-based Requirements Engineering?

1.2 Methodology

The research will be exploratory, because we want to learn more about the concept of crowd-based Requirements Engineering and because it is not certain what the outcome of the research will be. The exploratory research will be executed as a multiple-case study. This is because we want to look into more than one case in order to get a more information about Requirements Engineering. To answer the research question, there needs to be a collection of qualitative data. A qualitative research results in deeper insights of the concept than doing a quantitative research.

The data collection method is an interview. This research uses interviews for collecting data, because it will get more quality of data than for example a questionnaire. The interviews are conducted with organizations that use Requirements Engineering in their process of product development. There are 12 interviews executed. The questions of the interview are established beforehand. The questions are about the practices of the Requirements Engineering process and the effects of their approach. The interviews are recorded. These interviews are transcribed word for word. We will perform a thematic analysis. Once all the interviews were transcribed, there are given certain labels to the text fragments. This process is called coding. These labels represent the main topic of the text fragments. After labeling all the text fragments, they are compared with each other. Text fragments with a similar label are given an overarching label. This resulted in a list of codified terms and their frequencies. This labeled data is analyzed by searching for connections in the data. From the results the deliverables are made. These deliverables are explained in the next section.

1.3 Thesis overview

Section 2 discusses what Requirements Engineering is and the different types of Requirements Engineering. Crowd-based Requirements Engineering will be further explained; Section 3 describes the methodology used in this research. Section 4 describes the outcome from the qualitative research. Section 5 discusses the results and gives answer to the sub-questions. Section 6 concludes and gives answer to the research question. This bachelor thesis is performed at the University of Leiden and is supervised by T.D. Offerman MSc, P. van Leeuwen MBA and Dr. A.W. Laarman.

2 Literature Review

In this Literature Review we want to learn more about Requirements Engineering and the types of Requirements Engineering. We will first discuss what Requirements Engineering is. Then we will look into the process of Requirements Engineering. After having more knowledge on what Requirements Engineering is we will look into the different types of Requirements Engineering. In this section more focus will be put on crowd-based Requirements Engineering. As last, we will present some research discoveries of crowd-based Requirements Engineering. This Literature Review will be used to set up the interviews.

2.1 Requirements Engineering: What is it?

Requirements Engineering research already dates back to 1964[DB64]. In this year it was the first time Requirements Engineering was mentioned in literature. After this introduction of Requirements Engineering it was quiet for a little while. It wasn't until the mid 1970s that Requirements Engineering was acknowledged as a field on its own[Poh]. From then on more and more research has been done on Requirements Engineering. Although Requirements Engineering is gaining more interest, the research is not supported with experiences of Requirements Engineering in practice. This has been noticed several times in 1998 by researchers[BL98]. After this realization Requirements Engineering was more and more used in practice over time[CA07]. This way the research ideas and proposals could be evaluated. Additionally, Requirements Engineering only has become more important for professionals as well, as we can see in the number of professional publications.

To explain what Requirements Engineering is, we first need to understand what a requirement is. A requirement is defined as "A condition or capability needed by a user to solve a problem or achieve an objective."[LK95]. This definition has also been acknowledged by other researchers [Poh][Mac12].
Requirements Engineering is often defined as "The branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families."[Zav97]. This definition was first published by P. Zave in 1997. This definition is acknowledged by many researchers[Lap17][NE00].
The definition emphasises the importance of "real-world goals", the specifications and "evolution over time and across software families". "Real-world goals" show the motivation the world has for developing software systems. Specifications emphasise the activities of Requirements Engineering such as analyzing, validating, defining and verifying requirements. "Evolution over time" show that

Requirements Engineering will evolve in time because the world is constantly changing and growing. "Across software families" is mentioned to emphasise that the specifications can be used in other branches that are related to software engineering[NE00]. Requirements Engineering is covered by different literature streams, such as Enterprise Architecture[EW12], Software Engineering[WB13], Information systems[HS11] and Psychology[Mac17]. In this research we mainly apply and contribute to software engineering.

Requirements Engineering is the first phase of every product development. Requirements are "something that the customer needs"[Mac12]. It is the foundation of all products. It is important to have products that satisfy the customer in order to create competitive advantages and gain business growth. Requirements Engineering ensures that the needs of customers are defined and that they are clearly documented for the developers to understand. The process of Requirements Engineering helps preventing errors in the early stages. All this results in better customer satisfaction and higher product quality. Therefore, Requirements Engineering is still a hot topic nowadays.

2.2 The process of Requirements Engineering

If we look at the process of Requirements Engineering we can define this process as "the systematic process of developing requirements through an iterative co-operative process of analysing the problem, documenting the resulting observations in a variety of representation formats, and checking the accuracy of the understanding gained."[Poh93][Poh].

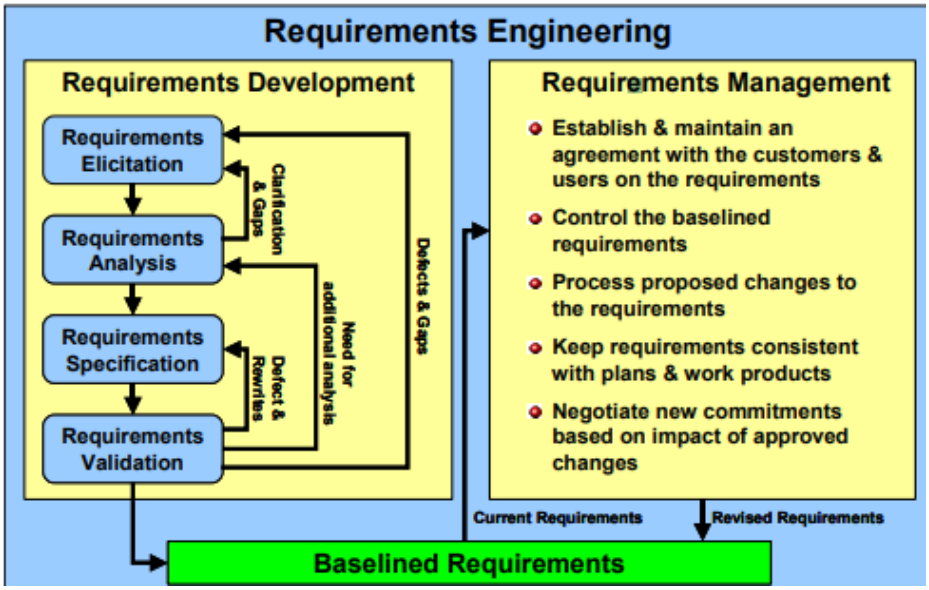

Figure 1: Process of Requirements Engineering[Wes05].

The main activities of Requirements Engineering are elicitation, analysis, specification, validation and management.These phases are shown in Figure 1 In the elicitation phase the development team want to figure out what the stakeholders want and need from the product. Stakeholders are the people who are interested and involved in the result of the product development. There are different elicitation techniques to find these needs of stakeholders such as questionnaires,

interviews or workshops. There are requirements derived from these elicitation techniques[Lap17]. In the analysis phase the requirements are represented in a model so they can be analyzed. The requirements are analyzed to ensure their clarity, feasibility, consistency and completeness[Wes05]. In the specification phase the requirements are represented clearly in a document so that it is understandable for the developers. In the validation phase the requirements will be reviewed by developers and stakeholders to check if the specification is correct and it meets the customers' needs[Lap17]. The requirements also need to be unambiguous, testable and traceable[Wes05]. In the management phase the requirements are managed over time. There may come new requirements to light which needs to be implemented[Lap17].

2.3 Types of Requirements Engineering

There are different types of Requirements Engineering: Traditional Requirements Engineering, Customer-specific Requirements Engineering, Market-driven Requirements Engineering and Crowd-based Requirements Engineering. The main difference of customer-specific Requirements Engineering with other types of Requirements Engineering is that the product development is done for one customer. Therefore the requirements consists only the needs of this customer. Market-driven Requirements Engineering differs mainly from the other Requirement Engineering types in that it has no defined users. Next to that, Market-driven Requirements Engineering has a short time-to-market. This is because there is a lot of competition in the market. You need to bring your product to the market on time, otherwise you lose customers. Crowd-based Requirements Engineering will be explained in the next section.

In Table 2.3 the differences between the types of Requirements Engineering are presented through some characteristics. The definition of every Requirements Engineering type is given. The characteristic 'stakeholders' refers to the people who are involved in the elicitation of requirements for the development of the product. The stakeholders' involvement defines in which phases of the development of the product the stakeholders are involved. For example this can be only during the elicitation phase, but also after the development of the product. The motivation describes in which way the stakeholders are motivated to be involved in the development of the product. Finally, the success measurement refers to how the success of the product is measured. Since there is not much known about Customer-specific Requirements Engineering for these characteristics in the literature, this type of Requirements Engineering will not be taken further into this research.

Characteristics	Traditional RE	Crowd-based RE	Market-driven RE	Customer-specific RE
Definition	"The systematic process of developing requirements through an iterative co-operative process of analysing the problem, documenting the resulting observations in a variety of representation formats, and checking the accuracy of the understanding gained."[Poh93][Poh]	"Crowd-based requirements engineering (Crowd-RE) is an umbrella term for automated or semi-automated approaches to gather and analyze information from a crowd to derive validated user requirements."[GSA+17].	"This type of software development is called market-driven and refers to the situation where the development costs of a generic product are divided among many buyers on an open market and where the potential profit is rewarded to the producer."[RB05].	Customer Specific Requirements Engineering is the same as traditional Requirements Engineering only is the product specifically made for the customer [RB05].
Stakeholders	Limited amount of representatives. These stakeholders are undefined [GSA+17].	Large group of current or potential users. These users are classified by their expertise and role [GSA+17].	Development organisations and set of undefined potential users [KDR+07].	The customer of the product[RB05].
Stakeholders' involvement	From elicitation until implementation [GSA+17].	Continuously involved, also after implementation [GSA+17].	Users are involved during elicitation and once the product is on the market, because users can give feedback [KDR+07].	No research data available.
Motivation	Not strongly focused on which way stakeholders are motivated [GSA+17].	Focused on motivating crowd-members with different approaches [GSA+17].	No research data available.	No research data available.
Success measurement	Stakeholders validate the requirements. They are responsible for the success of the product [Lap17].	The feedback of a product can be compared to feedback of similar products of different companies or to its older version to see if problems are resolved [GSA+17].	The gap between what the product delivers and what the customers wants is used to measure the success of the product [GGPT12].	No research data available.

The techniques used in each phase of Requirements Engineering are shown below for every Requirements Engineering type.

Elicitation techniques:
Traditional RE: Analyzing current product, think of requirements yourself, data mining, interview, introspection, observation, questionnaire, reuse, workshop[FGZ15].
Crowd-based RE: Eliciting general requirements with focus on creating personas for users, receiving run-time user feedback or elicitation tools for crowd[KLWA19]. Feedback channels[GSA+17], crowdsourcing platforms[VG20], User feedback through crowd-based monitoring[GSA+17].
Market-driven RE: Market survey, customer visits, internal sources, customers sending direct feedback or requirements[Per03], through suggestions, bug report and complaints of users [KDR+07].

Analysis techniques:
Traditional RE: Domain-driven development, formal specification, informal modelling, object-oriented analysis, prototyping, quality checks, structured analysis [FGZ15].
Crowd-based RE: User feedback through survey, functionality that detect overlapping requirements and dependency between requirement[VG20], textual data analysis, user behavioral data [KLWA19].
Market-driven RE: Discussion groups [Per03].

Prioritization techniques:
Traditional RE: Prioritizing, conflict management, handshaking, strategy alignment [FGZ15].
Crowd-based RE: Discussion system, user rating system, crowd-members voting[KLWA19], three-point-scale[VG20].
Market-driven RE: Time/cost estimates [Per03].

Specification techniques:
Traditional RE: Natural language, structured analysis diagrams, tables, UML diagrams, User screens[FGZ15].
Crowd-based RE: User story template, Use cases, users specify requirements. The crowd is not use much in this phase since users are not trained to specify requirements [VG20].
Market-driven RE: Natural language, Requirements are sometimes communicated within the development team instead of written down in documents[Per03].

Validation techniques:
Traditional RE: Automated checking, Inspection, peer review, prototype review, simulation, walk-through with stakeholders[FGZ15].
Crowd-based RE: Review of crowd-members, monitoring data on usage [VG20].
Market-driven RE: Selected customers test beta-versions of the product[Per03].

Management techniques:
Traditional RE: Versioning requirements, change management, traceability management [FGZ15].
Crowd-based RE: Stakeholder relation management [VG20], users' behaviors log [KLWA19], context and usage data from several sources[GSA+17].
Market-driven RE: Database to manage the constant flow of feedback[Per03].

2.3.1 What is crowd-based RE?

Crowd-based Requirements Engineering is an addition to the traditional Requirements Engineering. The main activities of Requirements Engineering are the same, but the people used in the process of gaining requirements are different. In crowd-based Requirements Engineering developers use a crowd which is a large group of current or potential users[GSA+17]. The crowd-members can be classified by their level of skill and their role [CA09]. The crowd is continuously involved in the process and after the implementation of the product. The crowd provides user feedback. This feedback can be software problems, improvement suggestions or new-product ideas. This feedback can lead to new requirements for the product. Crowd-based Requirements Engineering also uses crowd-based monitoring of software products to obtain new requirements. The data collection of these monitoring systems can be gathered from multiple sources. This provides context and usage data. This data can be used to better understand user feedback and create relevant requirements. The feedback that is gathered will be analyzed using automated analysis techniques such as text mining[GSA+17].
In crowd-based Requirements Engineering the crowd-members are categorized in different types. The types are derived from the motivation of crowd-members. Developers want crowd-members to have a genuine interest in the outcome of the product so that their participation is the most effective. To achieve this, the developers need to consider different types of approaches to motivate their crowd-members. Gamification is an example of a motivation technique[GSA+17].
Current research shows that there are two approaches seen in the literature of crowd-based Requirements Engineering; data-driven approaches and collaborative approaches. The data-driven approach is used to extract requirements from several data sources. User feedback, bug reports and other relevant data from these sources that can be extracted will be analyzed to create requirements. This approach is focused on the phase after the product development. The collaborative approach is used to develop the product together with the user. This can done by using tools such as web-based crowd-sourcing platforms. This way users can communicate with each other and make a prioritization together. This approach is focused in the phase previous to the product development. These approaches can also be used in combination[VG20].
The success of applying crowd-based Requirements Engineering can be measured through the user feedback. For example when in the feedback of a product a problem is mentioned, the next version of this product should get less feedback of this problem. Furthermore, the feedback of a product can be compared to feedback of similar products to see which product has the most positive feedback for example[GSA+17].

2.4 Research discoveries of crowd-based Requirements Engineering

In previous research has shown some discoveries in crowd-based Requirements Engineering. These discoveries are stated below.

- Based on research that is available it is stated that there is a positive relationship between users' involvement in the phase of Requirements Engineering and the products' success due to this involvement[BZ13].

- Previous research has shown that the crowd is used mostly in the elicitation phase and the analysis phase of the Requirements Engineering process. Studies use different approaches

in these activities to involve the crowd[KLWA19]. For example, in the elicitation phase the crowd can communicate their needs through a crowd sourcing platform. In the analysis phase the crowd fills in a survey for example to check the requirements on their quality and give feedback this way[VGS19] [VG20].

- Research has shown that Crowd-based Requirements Engineering has the following challenges: analyzing large amount of feedback(1), privacy of users(2), motivating the crowd(3), assuring quality of requirements(4), meeting users' expectations(5).

Challenge 1: In Crowd-based Requirements Engineering you receive a continuous stream of context and usage data. All this data needs to be analyzed. When the users are anonymous it is hard to seperate the users in subgroups. Smaller subgroups might be overlooked. Relevant requirements will be missed[GSA+17].
Challenge 2: Users must decide what level of privacy they want. This means that users can decide when, where and how feedback is collected. This must be implemented when building in monitors and channels for feedback[GSA+17].
Challenge 3: Motivating the crowd-members must not affect the users in a way that it will have effect on the usefulness and truthfulness of the feedback[GSA+17].
Challenge 4: Assuring the quality of requirements is a difficult challenge. You want to find out which circumstances and quality measurements are needed for the crowd to give useful input [KLWA19].
Challenge 5: Involving the users in the development process can give the users higher expectations of the product. The organization might not be able to meet these expectations[SDH+14].

Although the concept of crowd-based Requirements Engineering seems promising, many researchers have mentioned difficulties of the implementation and execution of crowd-based Requirements Engineering[SDH+14][KLWA19][GSA+17]. These complications raise questions about the effectiveness of crowd-based Requirements Engineering. It is stated that there is not much known about the successfully execution of crowd-based Requirements Engineering[GSA+17]. Therefore, it is difficult to state the precise effects of crowd-based Requirements Engineering. In this research we learn more about the concept of Crowd-based Requirements Engineering. We state the lessons learned of Crowd-based Requirements Engineering in practice. This way we contribute to resolving the gap in the literature of crowd-based Requirements Engineering in research and in practice.

3 Methodology

Crowd-based Requirements Engineering is a concept that is mentioned in literature, but there is not much known about the successful execution. Because there is so little known about crowd-based Requirements Engineering in practice, the research will first focus on how Requirements Engineering in general is implemented in practice. Later on there will be more focus on whether the process of Requirements Engineering in practice can be categorized as crowd-based Requirements Engineering. The next sections will describe the methodology of this research, the data collection method and the methodology for analyzing the data.

3.1 Exploratory, multiple-case, qualitative research

To answer the research question, there will be a qualitative, exploratory research methodology. The purpose of doing a qualitative research is to learn how processes are executed and why they are executed this way. You create understanding of people's experiences, behavior and beliefs[FHMD02]. It results in deeper insights of a concept than doing a quantitative research, because you receive depth of information. In contrast to quantitative research there is a small number of participants. The data of the qualitative research is words and therefore is interpretative [HHB20]. The research will be exploratory, because we want to learn more about the concept of crowd-based Requirements Engineering. The exploratory research will be executed as a multiple-case study. This is because we want to look into more than one case in order to get more information about Requirements Engineering. The goal for this qualitative, exploratory research is to get insights in the Requirements Engineering process of organizations.

3.2 Data collection method

The data collection method is an interview. This research uses interviews for collecting data, because it will get more quality of data than for example a questionnaire. When doing an interview you get insights in not only their opinion, but also in their experiences and behavior. Interacting with the participants helps avoiding misinterpretations of questions. It is also easier to understand the answers and their interpretation. Furthermore, an interview is more personal which makes it more reliable. The interviews will be semi-structured. Begin able to ask questions that are not established beforehand helps better understanding the answers of participants.

3.2.1 Interview

The interviews were conducted with organizations that use Requirements Engineering in their process of product development. This research is done with organizations that use Requirements Engineering to build software or an IT-related product. There is no further distinction made in organizations. At least 10 interviews were needed to be executed in order to get good insight with different points of view about the lessons learned from crowd-based Requirements Engineering. Finally there were 12 interviews executed. The questions of the interviews were established beforehand. The questions were formed after doing a literature review. In this literature review we reviewed several characteristics of crowd-based Requirements Engineering. The questions were based on these characteristics. The interview began with some introduction questions about the participants in order to get knowledge on the population of the research. In the second part of the interview the participants were asked to tell about how they performed certain steps of the Requirements Engineering process by giving an example. By letting them give an example, there was more context information received which helped better understand their process. In the last part the participants answered general questions about their Requirements Engineering process. To finish the interview we asked participants if they wanted to elaborate something or if they had any other questions and comments.

3.2.2 Participants

The participants work at different type of organizations and have different roles. All participants have knowledge on how Requirements Engineering is done in their organization. The participants are approached via email, Linked In and our social connections. The interviews were virtual. We wanted participants that are involved in the Requirements Engineering process. The Requirements Engineering process is always software related. The participants work at an organization that uses Requirements Engineering often. The participants often have a role as a business analyst, project manager, product owner, scrum master or system developer.

3.2.3 Interview procedure

After receiving an e-mail or text message of a participant that he or she is interested in my research, I tried to plan a meeting as soon as possible. I made sure the participants are informed about the fact that the interviews are recorded. Then the participants filled in a consent form of the University of Leiden. Once a convenient time and date for the interview was established, I sent them an invitation of the meeting through a virtual meeting platform of their choice. Most interviews are conducted with Microsoft Teams.

3.3 Data analysis

For the analysis of our data, we will be performing a thematic analysis. Thematic analysis is defined as "A method for systematically identifying, organizing, and offering insight into patterns of meaning (themes) across a data set." [BC12]. This method will provide meaning of data by finding commonalities. After this analysis, the research question can be answered. This method is flexible and accessible. It can be used to answer a variety of research questions and you don't need to be an expert to use this method. There are different ways to execute thematic analysis. There is an inductive and deductive approach. Inductive approach uses only the data to create themes. Deductive approach uses their own themes on the data to make interpretations. These approaches are mostly used together [BC12].

3.3.1 Analysis procedure

After doing an interview, it is transcribed word for word. The transcribed interviews are the data. First, the data is scanned. While scanning the data, notes are made to actively read the data. Second, labels are given to the data. This process is called coding. These labels represent the main topic of the text fragments. These labels are created using an combination of the inductive and deductive approach. After labeling all the data, they are compared with each other. Data with a similar label is given an overarching label. After creating a table with labels and the extracted data, these labels are reviewed. After reviewing the labels, the labels are defined. This results in a list of codified terms and their frequencies.

4 Results

In this section the results of the interviews are presented. First there will be a description of the participants and their characteristics. Then the results of the interviews will be given. This

includes the overall process of how Requirements Engineering is implemented in practice and the changes over time. Furthermore the characteristics of crowd-based Requirements Engineering will be presented. Next, the challenges and effects of the process of Requirements Engineering are given. And as last the participants' thoughts on crowd-based Requirements Engineering will be presented.

4.1 Participants

There are 12 interviews executed. The interviews have been virtually conducted due to the current COVID-19 situation. All the participants are Dutch and work in the Netherlands. The participants work at different type of organizations and have different roles in the organization. These organizations work with different product development methods. Participants have different roles in the Requirements Engineering process. They can be involved in the development of the product or can be leading the development team for example. All participants have knowledge on how Requirements Engineering is done in their organization. In the Table 1 the participants are shown with their roles in the organization.

Participant	**Type of organizations**	**Role in organization**	**Role in Requirements Engineering**
A	Bank	Business analyst	Involved in whole process
B	Software company	Technical director	Requirements gathering
C	Software company	Consultant	Feature lead/ Product owner
D	Commodity trading company	Tech lead	Facilitating
E	Business Intelligence Service	Business intelligence consultant	Involved in whole process
F	Consultancy firm	Junior consultant	Analyst
G	Engineering and Consultancy firm	Consultant	Requirements gathering
H	University	Project lead	Product Owner
I	Software company	Software developer	Software developer
J	Shipping & Transport	Continuous improvement manager	Involved in whole process
K	Energy utility company	Manager Customer IT	Manager IT teams
L	Energy utility company	IT solution designer	Involved in whole process

Table 1: The characteristics of the participants.

4.2 Requirements Engineering in practice

In this section the results are shown of how Requirements Engineering is implemented in practice by the participants of the interviews. The organizations use different product development methods. Some participants mentioned that they work Agile and use Scrum. Since Scrum is a method used in an Agile mindset, we will continue this research only referring to Agile. The participants work with Agile, Waterfall or a combination of both. In Table 2 the methods that are used are shown for the participants.

Product development methods	**Participants**
Agile	6
Waterfall	3
Agile & Waterfall	3

Table 2: Product development methods used by the participants.

To support the Requirements Engineering process the participants have supporting tools. As seen in the Graph 2 there are several tools used by the participants. Azure DevOps, Jira and Confluence are service management tools. For example, the requirements can be documented in a backlog which is accessible for every team member. Bitbucket is a code management tool which is used for the development and the deployment of a product. ITSM and Usabilla are tools to collect user feedback after product implementation. Figma and Uxpin are tools for making a prototype. Word is a tool for documenting text. Powerpoint is a tool to make presentations. Miro is a tool to collaborate online on a whiteboard.

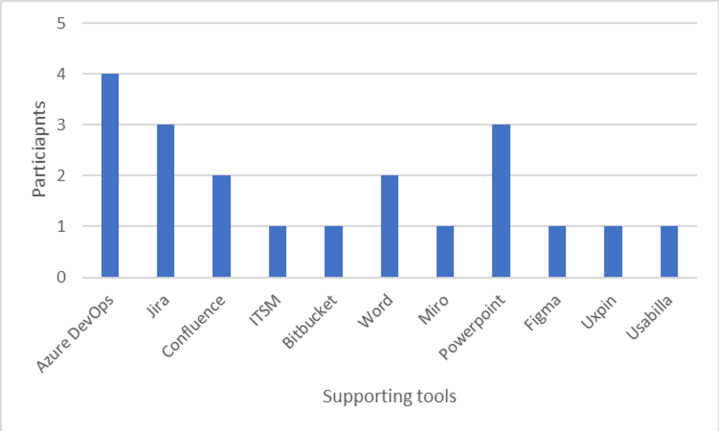

Figure 2: Quantified supporting tools by participants

After doing qualitative research there are 10 steps in the process of Requirements Engineering that arise. The discovered steps are shown in Table 3.

Steps	Agile	Waterfall	Agile & Waterfall	Total number of participants
Initiation of a need	6	0	3	9
Market research	3	0	0	3
Elicitation	6	3	3	12
Create overview Requirements	6	3	3	12
Analysis	6	3	3	12
Prioritization	6	3	3	12
Specification	6	2	3	11
Validation	6	3	3	12
Management	6	3	3	12
Elicitation after product implementation	5	1	2	8

Table 3: The number of participants shown for each Requirements Engineering in practice step distributed over the product development methods.

4.2.1 Initiation of a need

The start of the Requirements Engineering process begins with an initiation of a need for 9 out of the 12 participants. In this step a stakeholder or a user introduces the need for a new product. This initiation comes from the business itself for 3 out of 12 participants. And for 6 out of 12 participants the initiation comes for a customer. Participant C says for example "When a company comes in with the question 'Do you want to make software for us?', then we first go into a regular process to see 'What are they exactly asking? Can we make that?'." For the other participants it is not mentioned where the initiation of the process comes from.

Initiation of a need	Agile	Waterfall	Agile & Waterfall	Total number of participants
Organization itself	3	0	1	4
Customer	3	0	2	5

Table 4: Initiator of needs used by number of participants distributed over product development methods.

4.2.2 Market research

After getting the request for a new development or a new product 3 out of 12 participants will first do a market research. In this market research the participants consider outsourcing the development to an external company or hiring an external to do the development. They look into what is available in the market, what they can afford and what they are able to develop themselves. The reason for this market research is that it might be more cost efficient to hire an external company or that you sometimes might not have the resources to develop something. Participant D says for example "For example, if you had to choose between doing something manually and doing

something automated and the pain is small enough, then you can also hire someone for that. That might be more cost effective than having to build something again."

4.2.3 Elicitation

In this phase you want to gather requirements by figuring out what the customer wants and needs from the product. As you can see in Figure 3 there are different elicitation methods used in the Requirements Engineering process. The methods are also used in combination with each other. Interviews and workshops are the most popular method by the participants. In an interview the

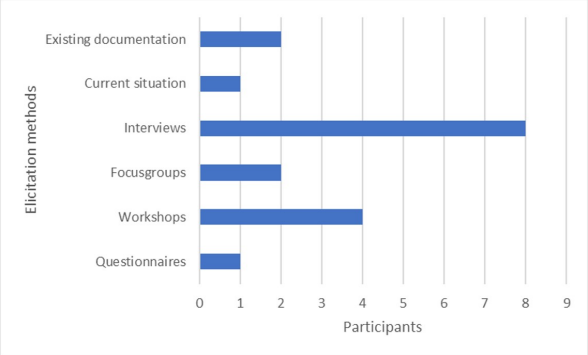

Figure 3: Quantified elicitation methods by the participants.

people ask questions to stakeholders such as end-users, product owner or the business department to figure out what they want from a product. This interview can be 1 on 1. In a workshop the facilitator does exercises or games to find what people need from a product. A workshop is performed with a group of people. To execute a workshop a session is first planned. There are multiple people joining this workshop which all need to be available. Participants C and F first conduct interviews to figure out the needs of the customer and then perform workshops to explore these needs further. For example, participant C says "And in the workshop, the graphical form is somewhat determined and how the application should manifest, how the flow is, things like that." Participant G conducts interviews in combination with looking at the current situation and the existing documents. Participant I also looks into existing documents besides performing interviews. Participant H uses questionnaires in combination with focus groups to figure out the needs of the end-user. In a focus group a group of people are asked about their opinion on the product. The questionnaire also asks some questions about the product to people, but this is a form that is sent online to potential users. They also perform interviews, but this is only with the IT department to figure out the technical requirements.

4.2.4 Document requirements

In this step the requirements are documented after the elicitation by all the participants. The flow of the product is represented by 3 out of 12 participants. The flow can be presented in a model or a flowchart. It gives a clear view of all the steps that are taken while using the product. The requirements are listed as text by 6 out of 12 participants. This can be a functional design or a business case or just a list of requirements. 5 out of 12 participants will create high level epics or even user stories to document the requirements. The epics consists of user stories. A user story is

written as: As a [role] I want [function], because [reason]. 1 participant out of the 5 participants that use user stories expands the user stories with definitions of done. Definitions of done are criteria that every user story should meet in order to complete the software development of the user story.

Document requirements	**Agile**	**Waterfall**	**Agile&Waterfall**	**Total participants**
Textual	2	2	2	6
User stories	3	1	1	5
Document flow	2	0	1	3
Definition of Done	1	0	0	1

Table 5: Documentation methods used by number of participants distributed over product development methods.

4.2.5 Analysis

The next phase is the analysis of requirements. In this phase the requirements are analyzed to ensure their clarity, feasibility, consistency and completeness. 8 out of 12 participants use conversations to analyze these requirements. These conversations can be with the business department or stakeholders to ask for further explanation. For example participant D says "And this is then presented to the rest of the group. And then they are allowed to continue discussing what is right and wrong about it." The analysis is also done by visualizations. 3 out of 12 participants use visualizations such as sketches or a prototype to analyze the requirements. It is mentioned that visualization helps bring the picture to live and result in different opinions then when they would not have seen the visualization. One participant has a checklist with some criteria that the requirements must meet. 2 out of 12 participants will analyze the requirements based on their experience. The requirements are analyzed by the development team itself. For example, participant C says "In general for the somewhat larger projects we do it more or less on a good feeling based on experience."

Analysis method	**Agile**	**Waterfall**	**Agile & Waterfall**	**Total number of participants**
Conversational	5	1	2	8
Visualization	2	0	1	3
Experience	0	1	1	2
Checklist	0	1	0	1

Table 6: Analysis methods used by number of participants distributed over product development methods.

4.2.6 Prioritization

In this step the requirements are prioritized. The requirements are prioritized by their added value to the product by 10 out of 12 participants. The requirement with the most added value to the product gets a higher priority. There are 2 participants that work with a minimum viable product to prioritize on the value of requirements. In this case the minimum requirements that are necessary for the product to work have the highest priority. There are 2 other participants that use the MoSCoW

method to make a prioritization. The MoSCoW method is used to let stakeholders give a category to a requirement. The categories vary from must-have, should-have, could-have and won't-have. The people who where also involved in the elicitation are asked in what category the requirements fit. Another participant creates a graph where the value of a requirements is plotted to the effort. This graph is created in a workshop with a minimum of 3 and a maximum of 7 representatives of groups of stakeholders. There are 3 participants out of 12 participants who prioritize by taking the sequence of the development of requirements into consideration. Some tasks or user stories need to be executed before other tasks.

Prioritization method	**Agile**	**Waterfall**	**Agile & Waterfall**	**Total number of participants**
Value	4	3	3	10
Sequence	2	0	1	3

Table 7: Prioritization methods used by number of participants distributed over product development methods.

4.2.7 Specification

In the specification phase the requirements are represented clearly in a document so that it is understandable for the developers. Although most participants have already documented the requirements in some way after the elicitation, some participants will also define them even more. 4 out of 12 participants use acceptance criteria to specify the requirements. In the acceptance criteria it is stated what specifications the requirement should meet. One participant also uses a definition of done to extension of this acceptance criteria. Participant D says for example "But usually they are put down as acceptance criteria like 'Well I have to meet here and here and only then it is finished'." 3 out of 12 participants use a functional design to specify the requirements. For example, participant E says "You can make a kind of functional design of what you eventually will technically realize." 7 out of 12 participants will specify the requirements as user stories. A user story is written as: As a [role] I want [function], because [reason]. 4 participants create sub-tasks under these user stories.

Specification method	**Agile**	**Waterfall**	**Agile & Waterfall**	**Total number of participants**
Acceptance criteria	1	0	3	4
Functional design	2	1	0	3
User stories	3	1	3	7

Table 8: Specification methods used by number of participants distributed over product development methods.

4.2.8 Validation

In the validation phase the requirements will be reviewed by developers and stakeholders to check if the specification is correct and it meets the customers' needs. In Table 9 the different validation methods are shown together with the number of participants that use that method for each product development method. 6 out of 12 participants review the requirements. This review can

be done within the development team or by asking stakeholders for feedback. 3 participants do these reviews with the product owner and the development team in sprint plannings or refinement sessions. Participant D mentions that the product owner has good insights most of the times to see if the requirements are still relevant. 5 out of 12 participants use visualizations to validate the requirements. This can be a demo, prototype or half of the product. For example, participant A says "And at the end of a sprint we always have demos and then we just show how it is currently." 4 out of 12 participants will do a test as a way of validation. This can be an acceptance test or a user test. In a user test users are asked to execute some exercises with the product and answer questions. In the acceptance test the product is tested by the initiator of the product. The initiator imitates the user experience while testing the product. For example, participant K says "The developer brings it to an acceptance environment. And the acceptance environment with us is identical to the production environment that the customer sees. So that's actually re-enacting the customer experience by whoever came up with it."

Methods	**Agile**	**Waterfall**	**Agile & Waterfall**	**Total number of participants**
Review	3	2	1	6
Visualization	3	1	1	5
Test	2	0	2	4

Table 9: Validation methods used by number of participants distributed over product development methods.

4.2.9 Management

Managing the requirements is done to make sure they are consistent and traceable and structured over time. 7 out of 12 participants use a tool to manage the requirements. In these tools it is clear what requirement is created, when and by who. 2 participants mention that the status of the requirement is also visible in this tool. 6 out of 12 participants have the requirements saved after product implementation, but the requirements are not used or managed in any way. 3 out of 12 participants make sure the requirements are visible for all stakeholders. 3 out of 12 participants only change the requirements when adjustments need to be made.

Management	**Agile**	**Waterfall**	**Agile&Waterfall**	**Total participants**
Tool	3	1	3	7
Accessible	3	1	1	5
Adjustments	1	2	0	3
After deployment: saved, not managed	3	2	1	6

Table 10: Management methods used by number of participants distributed over product development methods.

4.2.10 Elicitation after product implementation

This step is part of the managing phase of requirements. In this step the requirements are gathered after product implementation. This can be bug reports, product suggestions or ideas for a new product. 3 out of 12 participants do not do anything with the requirements after product implementation. There is not much feedback expected after finishing the product. There might be some technical issues, but those will be fixed as soon as possible. 3 out of 12 participants have a tool for collecting feedback of users. This tool is used for technical issues, new requests or to see what users think of the latest implementation. 3 out of 12 participants have someone in the organisation who keeps contact with the users. 2 out of 12 participants do user tests after product implementation. The users are then asked to perform an activity and show the way they do this activity.

Elicitation after product implementation	**Agile**	**Waterfall**	**Agile&Waterfall**	**Total number of participants**
Product finished	2	1	0	3
Tool for feedback	3	0	0	3
Contact	1	0	2	3
User tests	0	1	1	2

Table 11: Elicitation methods used by number of participants distributed over product development methods.

4.3 Changes in Requirements Engineering over time

The changes in the Requirements Engineering process can be divided in 4 categories. In the Table 12 the number of participants that belong to a certain category are shown.

Differences	**Agile**	**Waterfall**	**Agile & Waterfall**	**Total number of participants**
No Requirements Engineering	2	1	0	3
Product development method	3	0	2	5
More stakeholders involved	0	1	0	1
No changes	2	0	1	3

Table 12: The number of participants is shown for each category that represent the difference between the previous state and the current state of Requirements Engineering.

The category 'No Requirements Engineering' means that there was no Requirements Engineering process before. 3 out of 12 participants did not have a Requirements Engineering process before they showed up.
The category 'Product development method' refers to the previous product development process used by the participants. 4 participants used waterfall as a previous product development method.

2 participants mention that this brought a lot of problems. For example participant L says "And six months later you started talking to users again, then the product was ready, and it could be tested. That caused so many problems." One participant used to work in a project from. In the project form they made schedules and plannings. In the current situation they work in biweekly cycles. The time given is much shorter. In the current situation these participants work more Agile.
The category 'More stakeholders involved' means that in the previous situation some stakeholders were not taken into account. Participant F mentioned that they only interviewed the project leader to elicit requirements. In the current situation there are more stakeholders also involved in the process.
The category 'No changes' means that the process of Requirements Engineering did not change. 3 out of 12 participants did not have changes in their process of Requirements Engineering.
These Requirements Engineering processes that did undergo changes also have an effect. In the Table 13 the number of participants is shown for certain effects that they have seen.

Effects	**Agile**	**Waterfall**	**Agile & Waterfall**	**Total number of participants**
More communication	2	1	2	5
Faster results	2	1	1	4
Faster adjustments	2	0	0	2

Table 13: Effects seen by number of participants distributed over product development methods.

There is more communication seen in the process of Requirements Engineering. 3 out of 12 participants communicate more with the stakeholders. Participant K says "In fact, the requirement of every user story is that you ask for feedback from the customer as soon as possible, because only then you will know whether you are doing the right thing." 3 out of 12 participants have more communication within the development team itself. The knowledge is more shared within the team and everybody works together as a team on the project. 4 out of 12 participants mentioned that they have faster results. In the current situation the participants can show results to their stakeholders faster. 2 out of 12 participants mention that adjustments can be made faster. Sometimes customers want more features later on in the process. In this case adjustment can be made easily. For example, participant J says "And now if there are any questions then we can endure it. And if there are any changes that need to be made, then we can adjust it relatively easily and quickly."

4.4 Characteristics of crowd-based Requirements Engineering

Crowd-based Requirements Engineering can be distinguished from other types of Requirements Engineering by user involvement, motivation and success measurement. In this section the results are shown of how these characteristics are performed in practice.

4.4.1 User involvement

The users are mostly involved in the elicitation of requirements as seen in Table 14. In most cases there are a few users involved in the elicitation of requirements. The number of users are between 1 and 20. 4 out of 12 participants only have one person who is involved. This can be a end-user representative or someone from the business or the customer itself. It is mentioned by

2 participants that not all the opinions of users are wanted, because that would simply be too many. One participant mentioned that they wanted to involve a representative of every group of users. Another participants mentioned that it is important to involve every user. There is one exception where 1800 users were involved in a questionnaire, but in the other elicitation practices a group of 15-20 users was used. In the analysis of requirements 6 out of 12 participants involve the same users as used in the elicitation phase. The other participants analyze the requirements with development team and sometimes involve the business department or a representative of the users. In the prioritization of requirements 4 out of 12 participants involve the user in this phase. These users are the same as the users from elicitation. In the specification phase there are no users involved. In the validation phase 6 out of 12 participants involve the users. These users are the same as the users from elicitation. In the management of requirements there are no users involved. In the elicitation of requirements after product implementation 8 out of 12 participants mention that the users of the product can give feedback.

Requirements Engineering phases	**Participants**
Elicitation	10
Analysis	6
Prioritization	4
Specification	0
Validation	6
After product implementation	8

Table 14: Quantified involvement of users shown for each phase of Requirements Engineering.

4.4.2 Motivation

Stakeholders or users that are involved in the elicitation, analysis and validation can be motivated in several ways. 'Intrinsic motivation' means that participants do not need to motivate the stakeholders or users that are involved, because they are automatically motivated by being a part of the process. 4 out of 12 participants use 'motivation techniques' such as explaining the advantages of their involvement to stakeholders or being very positive. 2 out of 12 participants motivate stakeholders or users by keeping the people involved through the process.

Motivation methods	**Agile**	**Waterfall**	**Agile&Waterfall**	**Total number of participants**
Intrinsic motivation	1	2	1	4
Involvement of process	2	0	0	2
Motivation techniques	2	1	1	4

Table 15: Quantified motivation methods by participants distributed over product development methods.

4.4.3 Success measurement

Once the product is implemented and used by the actual users, there are ways to measure the products' success. In the Graph 16 the different success measurement are shown.

Methods	Agile	Waterfall	Agile&Waterfall	Total number of participants
Meet products' goals	1	1	1	3
Users vs errors	1	0	0	1
User satisfaction	3	0	2	5
Usage	3	1	0	4
User tests	0	1	0	1

Table 16: Quantified success measurements by participants distributed over product development methods.

The success measurement categorized as 'Meet products' goals' refers to whether a product is successfully developed for its purpose. This measurement is used by 3 out of 12 participants. The purpose of every product can be different. For example, participant D says "In our business you have to either be faster or smarter than your competitors. In the end there will also be looked at like 'What we've developed, it must be running a bit fast'. And if it doesn't, then it has to go back to the drawing board. Then it doesn't suffice."
One participant measures the success of a product by looking into the amount of users versus the amount of errors a product has. This ratio is then compared to a standard users versus errors ratio. If the ratio is higher, then the product is not successful.
The user satisfaction is measured by 5 out of 12 participants. This can be done by a score that shows the customer satisfaction of the user. For example, participants K says "We measure what a particular feature has done through the User Bella. So in user Bella, people can also rate from zero to ten 'what did you think of this?'."
The success measurement categorized as 'Usage' refers to the whether the product is used by the users. If the product is used, then that means that it is a success. For example, participant D says "Well eventually whether it is used. It is of course important that you do not make something that is not used. That's the most important."
The success is also measure by user tests. In the user tests all the requirements that were set up in the beginning of the process are validated. For example, a requirement can be that you must be able the easily log in. By testing the product with the user, you can see if this requirement is successfully executed.

4.5 Challenges and effects of Requirements Engineering

The process of Requirements Engineering has some effects and challenges.These will be explained in the following section.

4.5.1 Challenges

In the whole process of Requirements Engineering challenges may arise. In the Table 17 the different challenges are shown.

Challenges	Agile	Waterfall	Agile&Waterfall	Total number of participants
Uncooperative developers	0	1	0	1
Different goals	1	1	0	2
Expectation management	1	0	1	2
Complete requirements	1	1	1	3
Motivate stakeholders	1	1	2	4

Table 17: Quantified challenges by participants distributed over product development methods.

The challenge to motivate stakeholders refers to stakeholders that are not motivated to be involved in the process or get demotivated along the way. 4 out of 12 participants run into this challenge. For example, participant D says "I think you have the biggest challenge indeed to get people motivated like 'Deliver your input and make sure that if you have an opinion about something that you air it'."
The challenge categorized as 'Complete requirements' refers to the difficulty to collect all the requirements that are necessary. It is difficult to know when your requirements are complete. Stakeholders might have implicit assumptions or they find it difficult to clearly state what they want.
'Uncooperative developers' is a challenge where developers do not want to perform Requirements Engineering beforehand, because they think that they already know the needs. Participants G says "That there is little attention, little time for it. Like 'Go on now build or something'."
The challenged categorized as 'Expectation management' refers to dealing with the functional wishes and the technical possibilities. When stakeholder deliver requirements they expect to see them back in the end product. However, it is not always possible from a technical perspective. This results in disappointment from the stakeholders. For example participant H says "Some things are not possible or are very complicated or are possible but take a lot of time. What do you do then? Do you adjust your requirements or do you stay on the same path any way?"
The last challenge is 'different goals'. This can be between the development team or even the users who have different objectives. For example participant K says "What I think is very good, you might find garbage."

4.5.2 Effects

There are different effects of the process of Requirements Engineering. These effects are shown in Table 20 with the number of participants distributed over the product development methods.
The first effect is the success of the product. 4 out 12 participants mention that they receive a successful product after development. This results in content customers or users. For example, participant E says "And there are organizations that are at a low level with certain business intelligence things. And if you take it to a higher level in 1 go, they actually skip some stages, then the people can be of course very happy."
There is also effect on the process itself. The process is for 4 out of 12 participants more efficient. This means that the development is easier and faster. Developers can faster show results to stakeholder to get feedback. For example participant C says "So the way of working saves a lot of time. So we can faster start developing, before we have the requirement all the way to a certain point filtered out."

Effects	Agile	Waterfall	Agile & Waterfall	Total number of participants
Success of product	2	0	2	4
Process efficiency	2	1	1	4
Better involvement	2	1	0	3
Better understanding requirements	0	2	0	2
Disappointment	0	0	1	1
Meet expectations	1	0	0	1
Knowlegde of organization	1	0	0	1

Table 18: Effects seen by number of participants distributed over product development methods.

The effect 'Better involvement' refers to the involvement of stakeholders or users that is more meaningful. The involvement genuinely influences the outcome of the product. Participant J says for example "So being involved creates a certain kind of support base that makes it not like 'Hey we deliver something and it disappears dusty in a drawer'."
The effect 'Better understanding requirements' refers to better understanding the purpose behind a requirement and therefore it will better correspond to the stakeholders wishes. One participant also mentioned disappointment as an effect. This has to do with expectation management. When you involve people in the process, they will get expectations of the end product. Another participants mentioned the effect that they always meet expectations of the customer. And the last effect is also mentioned by one participant. This effect is that the knowledge of what happens in the organization is increased for each individual in the development team.

4.6 Crowd-based Requirements Engineering

In this section the results of the opinions of the participants are shown. The participants estimated how crowd-based their our Requirements Engineering process is. In the Table 19 the number of participants are shown for the two given estimations. The participants are also distributed on their product development method. As seen in the Table 7 out of 12 participants estimate their Requirements Engineering process as not crowd-based. 5 out of 12 participants mentioned that they have some aspects of crowd-based, but don't believe their process is totally crowd-based.

Crowd-based estimation	Agile	Waterfall	Agile & Waterfall	Total number of participants
Not crowd-based	3	2	2	7
Some aspects of crowd-based	3	1	1	5

Table 19: The number of participants are shown for each estimation on how crowd-based their Requirements Engineering process is.

The main reason given for not being crowd-based or for not being totally crowd-based is that the

participants don't use a crowd. They use a small group of stakeholders or users which cannot be called a crowd. This reason is given by 8 out of 12 participants. For example, participant B says "Our crowd is of course not very big, because most of our organizations are just not that big.". One participant believes their process is not totally crowd-based, because the crowd can only give feedback when they're asked. So the crowd is not able to actively approach the developer with their feedback. Another participant gives the reason that the crowd is not continuously involved.
The main reason given for having some aspects of crowd-based is that the participants do involve some sort of crowd. This reason is given by 4 out of 12 participants. For example, participant F says "But I think because we have all kinds of stakeholders involved in the project that we're already moving in that direction." One participant believes that they're partially crowd-based because they also create an environment where you stimulate people to give feedback.

4.6.1 Perceived effects of crowd-based Requirements Engineering

The participants all have shared their thoughts on the concept on crowd-based. In Table 20 it is shown what effects the participants see or what opinion they have. 3 out of 12 participants mention that they believe you will receive a better product when doing crowd-based Requirement Engineering. For example, participant K says "Well, the big advantage of what I just outlined is the sooner you bring it down to a certain population, the better your product is that you develop. And in the end, that's just the most important thing. So I definitely see the point." 2 participants mention that monitoring systems can help understand how users use the product. Participant F says for example "I just mentioned user experience. If you see from such a requirement 'Is it really used like that by the user?' that you can get a much clearer with such a monitoring system. So I think it will help a lot." This participant also mentions that it might be a lot of work. But he also believes that crowd-based Requirements Engineering can result in saving time and money, because you have a better preparation on the development. 5 out of 12 participants believe that crowd-based Requirements Engineering is suitable for bigger projects. These projects have more users. For example, participant G says "I can imagine that you want to build something new for the worldwide and that you give all kinds of people the opportunity to contribute." One out of 12 participants believes that crowd-based Requirements Engineering cannot be executed in a Agile mindset. The participants says "You know, actually that conflicts with the Agile/Scrum idea, where you basically just take some spokespersons of the stakeholders from the crowd and try to achieve results as quickly as possible." One out of 12 participants mentions that the needs might be unclear. The participant says "One might have a very good comment or wish and explains it well. And the other says it in a way that understand nothing at all."

Crowd-based recommendation	**Agile**	**Waterfall**	**Agile & Waterfall**	**Total number of participants**
Better product	1	1	1	3
Possible with bigger project	3	1	1	5
Monitoring benefit	0	2	0	2
Save time	0	1	0	1
Not with Agile	1	0	0	1
Unclear needs	1	0	0	1

Table 20: The perceived effects quantified by participants distributed over the product development methods.

5 Discussion and Further Research

This section will discuss the sub-questions mentioned in the Introduction. The three sub-questions are listed below.

- How is Requirements Engineering implemented in practice?
- How is the process of Requirements Engineering transformed over time?
- What are the perceived effects of using more crowd-based Requirements Engineering?

5.1 Steps of Requirements Engineering in practice

Literature has shown that the process of traditional Requirements Engineering has five phases; Elicitation, Analysis, Specification, Validation and Management[Lap17]. All these phases are also seen in the results of this research. Therefore we can confirm that these steps in general always exist in the process of Requirements Engineering in practice. The techniques in these five phases used by participants are in line with the traditional Requirements Engineering techniques mentioned in previous studies[FGZ15]. Additionally, we have found some other steps.

The first step we found is the initiation of a need. In this step a stakeholder or a user introduces the need for a new product. Although not all the participants have mentioned it, it is mentioned by the majority of the participants. Therefore it is likely that this step is part of the Requirements Engineering process in practice.

The next step we have seen in the results is a market research. This is not often mentioned and we have not seen it as a step of Requirements Engineering in literature. The market research is usually done when an organization doesn't have enough resources to develop a product. Therefore this step does not seem necessary in the Requirements Engineering process in practice.

The next step which did not occur in the traditional phases of Requirements Engineering is documenting the requirements after elicitation. It is interesting that this step is not mentioned in

literature, since this step is done by all the participants. After the elicitation the requirements are documented in a way that they have an overview of the results of the elicitation. The specification of requirements happens in traditional Requirements Engineering after the analysis. Apparently in these cases this specification of requirements is done earlier in the process of Requirements Engineering in a less detailed way. This might be because this overview of requirements is needed to get feedback on them in the analysis phase.

Therefore we can answer to the sub-question 'How is Requirements Engineering implemented in practice?' that it is implemented through the steps seen in Figure 4 and that there are traditional Requirements Engineering techniques used in these steps.

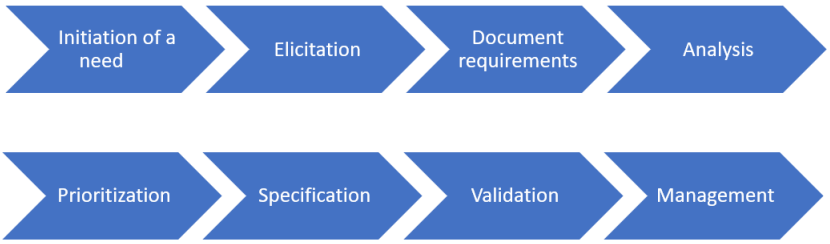

Figure 4: Steps of Requirements Engineering in practice.

5.2 Towards crowd-based Requirements Engineering

Although the Requirements Engineering processes in practice in the results cannot be categorized as crowd-based Requirements Engineering, some participants already have some aspects of crowd-based Requirements Engineering. These aspects will be discussed further in this section.

5.2.1 Involvement of users in Requirements Engineering in practice

Although it is stated in research that the crowd is mostly used in the elicitation phase and the analysis phase, we see in our results that users are involved also in the validation phase and after product implementation[KLWA19]. This matches the aspect of crowd-based Requirements Engineering where users are continuously involved in the process of Requirements Engineering[GSA+17]. Although the participants did not give this characteristic as a reason for being crowd-based, some of them apparently do have this aspect of crowd-based Requirements Engineering in their Requirements Engineering process in practice. This might be because the participants have made the transition to an Agile model of Requirements Engineering. There is a big transition seen over time in the results from the Waterfall model to the Agile model in Requirements Engineering in practice. This change has also been mentioned by other studies[Kas14][VC17]. In previous research it is stated that in an Agile model of Requirements Engineering the stakeholders are more involved[Kas14]. Therefore this could be the reason for the participants having the stakeholders more involved. Additionally we can therefore answer to the sub-question 'How is the process of Requirements Engineering transformed over time?' that for some of these participants there has been a transformation from using the Waterfall model to the Agile model and that there is more involvement of stakeholders or users. Since crowd-based Requirements Engineering is mainly focused on involving more users

and involving them continuously[GSA+17], it would be interesting for further research to find out if crowd-based Requirements Engineering can add more value to an Requirements Engineering process with an Agile model by involving more users in the process.

A perceived effect of crowd-based Requirements Engineering mentioned by the participants is a better product. This reason is given as an effect of more involvement of stakeholders or users. It has also been stated in research that their is a positive relation between the users' involvement and the products' success[BZ13]. Therefore this result is in line with previous research. On the other hand, it is also mentioned that involving more stakeholders or users results in expectations from these stakeholders or users that might not be met in the development of the product due to technical difficulties. This results in disappointment from stakeholders. This challenge is also seen in previous research[SDH+14]. It would be interesting for further research to find out what techniques are used by organizations to manage these expectations.

Another perceived effect is the benefit of monitoring systems. These monitoring systems collect context- and usage data. It is mentioned in the results that it helps better understand the users. This is in line with previous research where it is stated that monitoring systems can be used to better understand the feedback of users[GSA+17]. Therefore we can answer to the sub-question 'What are the perceived effects of using more crowd-based Requirements Engineering?' that the perceived effects could be a better product, more expectations from stakeholders or users and a better understanding of users' feedback according to the participants.

The number of users involved in the process of Requirements Engineering differs. The range of users are from 1 to 20 with one exception of 1800. But these number of users who are involved don't really give us enough information. In Table 2.3 from the Literature Review the crowd is defined as 'Large group of current or potential users'[GSA+17]. An organization could have a total of 100 users of which 50 users can be considered a crowd. But if you have an organization with millions of users, 50 users doesn't seem to be a large group of these users. Therefore we cannot give an indication whether a participant involves a small or a large part of the total users, because we don't know the total number of users of each organization. This number of total users is different for every organization and depends on the complexity of the product. Therefore, it would be interesting for further research to find out what percentage of users is involved in the Requirements Engineering process in practice. Additionally, this definition of a crowd in Table 2.3 is not clear. An adjustment can be made in this Table. The stakeholders should be defines as for example 'Group of current or potential users which is 50 percent of the total users'.

5.2.2 Motivation

There are 6 participants that have mentioned that they use motivation techniques to motivate users or stakeholders to give good input. However, we don't know whether these techniques result in intrinsic motivation. A study from Eduard C. Groen, 'The crowd in requirements engineering: The landscape and challenges', mentions that crowd members can be assigned to different categories by their attitude and motivation[GSA+17]. It would be interesting for further research to find out what motivation techniques are used for these different categories.

5.3 Limitations

The limitation of this research is that the sample size is small. It would be interesting to interview a larger group of people to confirm these findings. Another limitation is that some participants don't have a very long experience with Requirements Engineering. Therefore, there might be less changes mentioned. It would be interesting to interview these participants in a year or even two years to find out how their process is evolved. The changes over time could be stated more precisely.

6 Conclusions

After reviewing the Requirements Engineering process in practice and the aspects of crowd-based Requirements Engineering in these processes we answer the research question: "What are the lessons learned of using more crowd-based Requirements Engineering?". The lessons learned about crowd-based Requirements Engineering are that for these participants there isn't any awareness for the concept of crowd-based Requirements Engineering in practice. Additionally, it seems that the Requirements Engineering process is not keeping up with the digitization of the rest of the world for these participants. If we zoom in on the user involvement in crowd-based Requirements Engineering, we see that the stakeholders or users unconsciously continuously are involved by the participants. It is mentioned by participants that more involvement of stakeholders or users can result in a better product and better understanding users' feedback. The lesson learned in this is that there might be expectations of these stakeholders or users which need to be managed.

We can conclude for these participants that there isn't any awareness of the concept of crowd-based Requirements Engineering, because none of the participants perform this type of Requirements Engineering in practice. Additionally, the participants that do have some aspects of crowd-based Requirements Engineering did not consciously implement these aspects to become more crowd-based. Therefore the participants are not aware of this concept.

Although the world is becoming more digital and more intelligent, the participants have not mentioned any transitions over time to innovate the Requirements Engineering process. Crowd-based Requirements Engineering mentions monitoring systems that can be used to collect context- and usage data. Furthermore, linguistic analysis techniques such as text mining can be used to analyze this data[GSA+17]. Although this way the Requirements Engineering process could be more digital, we haven't seen any initiative towards this digital requirements gathering technique. Therefore the Requirements Engineering process of the participants is not keeping up with the digitization.

The high involvement of stakeholders or users can result in disappointment, because they create expectations while contributing in the product development. It is mentioned by the participants that although a requirement can have high value for the product, it sometimes cannot be implemented because of technical limitations. This can create disappointment and stakeholders or users can get demotivated. Therefore it is stated that expectation management should be considered when stakeholders or users are highly involved.

It would be interesting for further research to interview more participants to confirm that there isn't awareness for the concept of crowd-based Requirements Engineering and that expectations of

users' should be managed with highly involved users.

References

[BC12] Virginia Braun and Victoria Clarke. Thematic analysis. 2012.

[BL98] Daniel M Berry and Brian Lawrence. Requirements engineering. *IEEE software*, 15(2):26–29, 1998.

[BZ13] Muneera Bano and Didar Zowghi. Users' involvement in requirements engineering and system success. In *2013 3rd International Workshop on Empirical Requirements Engineering (EmpiRE)*, pages 24–31. IEEE, 2013.

[CA07] Betty HC Cheng and Joanne M Atlee. Research directions in requirements engineering. In *Future of Software Engineering (FOSE'07)*, pages 285–303. IEEE, 2007.

[CA09] Betty HC Cheng and Joanne M Atlee. Current and future research directions in requirements engineering. In *Design requirements engineering: A ten-year perspective*, pages 11–43. Springer, 2009.

[DB64] J Dresner and KH Borchers. Maintenance, maintainability, and system requirements engineering. Technical report, SAE Technical Paper, 1964.

[EW12] Wilco Engelsman and Roel Wieringa. Goal-oriented requirements engineering and enterprise architecture: Two case studies and some lessons learned. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pages 306–320. Springer, 2012.

[FGZ15] Samuel A Fricker, Rainer Grau, and Adrian Zwingli. Requirements engineering: best practice. In *Requirements Engineering for Digital Health*, pages 25–46. Springer, 2015.

[FHMD02] Ellie Fossey, Carol Harvey, Fiona McDermott, and Larry Davidson. Understanding and evaluating qualitative research. *Australian & New Zealand Journal of Psychiatry*, 36(6):717–732, 2002.

[GDA15] Eduard C Groen, Joerg Doerr, and Sebastian Adam. Towards crowd-based requirements engineering a research preview. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pages 247–253. Springer, 2015.

[GGPT12] Tony Gorschek, Andrigo Gomes, Andreas Pettersson, and Richard Torkar. Introduction of a process maturity model for market-driven product management and requirements engineering. *Journal of software: Evolution and Process*, 24(1):83–113, 2012.

[GSA+17] Eduard C Groen, Norbert Seyff, Raian Ali, Fabiano Dalpiaz, Joerg Doerr, Emitza Guzman, Mahmood Hosseini, Jordi Marco, Marc Oriol, Anna Perini, et al. The crowd in requirements engineering: The landscape and challenges. *IEEE software*, 34(2):44–52, 2017.

[HHB20] Monique Hennink, Inge Hutter, and Ajay Bailey. *Qualitative research methods*. Sage, 2020.

[HS11] Jonny Holmström and Steven Sawyer. Requirements engineering blinders: exploring information systems developers' black-boxing of the emergent character of requirements. *European Journal of Information Systems*, 20(1):34–47, 2011.

[Kas14] Mohamad Kassab. An empirical study on the requirements engineering practices for agile software development. In *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications*, pages 254–261. IEEE, 2014.

[KDR+07] Lena Karlsson, Åsa G Dahlstedt, Björn Regnell, Johan Natt och Dag, and Anne Persson. Requirements engineering challenges in market-driven software development–an interview study with practitioners. *Information and Software technology*, 49(6):588–604, 2007.

[KLWA19] Javed Ali Khan, Lin Liu, Lijie Wen, and Raian Ali. Crowd intelligence in requirements engineering: Current status and future directions. In *International working conference on requirements engineering: Foundation for software quality*, pages 245–261. Springer, 2019.

[Lap17] Phillip A Laplante. *Requirements engineering for software and systems*. CRC Press, 2017.

[LK95] Pericles Loucopoulos and Vassilios Karakostas. *System requirements engineering*. McGraw-Hill, Inc., 1995.

[Mac12] Linda A Macaulay. *Requirements engineering*. Springer Science & Business Media, 2012.

[Mac17] Iain S MacLeod. 4 the practice of engineering psychology-a uk approach. *Engineering Psychology and Cognitive Ergonomics: Volume 5: Aerospace and Transportation Systems*, page 31, 2017.

[MAS17] Pradeep K Murukannaiah, Nirav Ajmeri, and Munindar P Singh. Toward automating crowd re. In *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pages 512–515. IEEE, 2017.

[NE00] Bashar Nuseibeh and Steve Easterbrook. Requirements engineering: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering*, pages 35–46, 2000.

[Per03] Anne Persson. Market-driven requirements engineering processes for software products-a report on current practices. 2003.

[Poh] Klaus Pohl. *Requirements engineering: An overview*.

[Poh93] Klaus Pohl. The three dimensions of requirements engineering. In *International Conference on Advanced Information Systems Engineering*, pages 275–292. Springer, 1993.

[RB05] Björn Regnell and Sjaak Brinkkemper. Market-driven requirements engineering for software products. In *Engineering and managing software requirements*, pages 287–308. Springer, 2005.

[SDH+14] Remco Snijders, Fabiano Dalpiaz, Mahmood Hosseini, Alimohammad Shahri, and Raian Ali. Crowd-centric requirements engineering. In *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*, pages 614–615. IEEE, 2014.

[VC17] Wilfred Van Casteren. The waterfall model and the agile methodologies: A comparison by project characteristics. *Research Gate*, 2:1–6, 2017.

[VG20] Pascal Vogel and Christian Grotherr. Collaborating with the crowd for software requirements engineering: A literature review. 2020.

[VGS19] Pascal Vogel, Christian Grotherr, and Martin Semmann. Leveraging the internal crowd for continuous requirements engineering-lessons learned from a design science research project. 2019.

[WB13] Karl Wiegers and Joy Beatty. *Software requirements*. Pearson Education, 2013.

[Wes05] Linda Westfall. Software requirements engineering: what, why, who, when, and how. *Software Quality Professional*, 7(4):17, 2005.

[Zav97] Pamela Zave. Classification of research efforts in requirements engineering. *ACM Computing Surveys (CSUR)*, 29(4):315–321, 1997.

A Appendix

A.1 Interview guide

Thank you for participating in this interview. My name is Louise Kubatz. I am 21 years old and I am studying the Bachelor Informatica and Economie at the University of Leiden. I am currently writing my bachelor thesis that is about Requirements Engineering. I want to evaluate the process of Requirements Engineering in practice and discover the effects. I will use this to make a list of recommendations. I want to do these interviews with people that work at companies where their software is directly used by customers through for example an application or at companies where their software is outsourced to clients. Companies that use Requirements Engineering iteratively are ideal to do this interview with. Therefore I was looking forward to interview you about this process. I want to interview people who are in control or part of the execution of the Requirements Engineering process. This will often be people that have a role as a business analyst, project manager, product owner, scrum master or system developers. The interview will be approximately 60 minutes. I will be recording this interview. I will be analyzing and thereby using the data of this interview. Before starting the interview I want to confirm that you give permission for recording, analyzing and using the data from this interview.

Introduction questions:

What is your name?
What is the name of the company you work for?
Can you shortly describe what your company does?
What is your role in the company?
What is your role in the process of Requirements Engineering?
What software product is developed after your Requirements Engineering process?
In what context is Requirements Engineering typically performed at your company?

Interview questions: I will be asking questions about each step of the process.

Elicitation:
The first question is about the elicitation phase. In this phase you want to get knowledge on the needs of the customers. Requirements are created from these needs.

Can you tell me by giving an example how requirements typically are elicited?

Guiding questions:
Who is involved in the elicitation of requirements?
How much end-users are involved in this part of the process?
What techniques are used in the elicitation phase?
Agile/Scrum: Is this done in every sprint or are there certain sprints in where this is done?

Analysis:
The next question is about the analysis of requirements. In this phase the requirements are analyzed on their clarity, feasibility, consistency and completeness.

Can you tell me by using the same example how the requirements typically are analyzed?
If not, can you tell me by giving another example how the requirements typically are analyzed?

Guiding questions:
Who is involved in the analysis of requirements?
How much end-users are involved in this part of the process?
What techniques are used in the analysis phase?
Agile/Scrum: Is this done in every sprint or are there certain sprints in where this is done?

Prioritization:
The next question is about the prioritization of requirements.

Can you tell me by using the same example how the requirements typically are prioritized?
If not, can you tell me by giving another example how the requirements typically are prioritized?

Guiding questions:
Who is involved in the prioritization of requirements?
How much end-users are involved in this part of the process?
What is in general the size of the group of people that are involved?

What techniques are used in the prioritization phase?

Specification:
The next question is about the specification of requirements. In the specification phase the requirements are represented clearly in a document so that it is understandable for the developers.

Can you tell me by using the same example how the requirements typically are specified?
If not, can you tell me by giving another example how the requirements typically are specified?

Guiding questions:
Who is involved in the specification of requirements?
How much end-users are involved in this part of the process?
What techniques are used in the specification phase?
Agile/Scrum: Is this done in every sprint or are there certain sprints in where this is done?

Validation:
In the validation phase the requirements will be reviewed by developers and stakeholders to check if the specification is correct and it meets the customers' needs.

Can you tell me by using the same example how the requirements typically are validated?
If not, can you tell me by giving another example how the requirements typically are validated?

Guiding questions:
Who is involved in the validation of requirements?
How much end-users are involved in this part of the process?
What techniques are used in the validation phase?
Agile/Scrum: Is this done in every sprint or are there certain sprints in where this is done?

Management:
To manage requirements they need to be structured over time. The requirements need to be consistent and traceable after changes are made.

Can you tell me by using the same example how the requirements typically are managed?
If not, can you tell me by giving another example how the requirements typically are managed?
Can you tell me by using the same example how the requirements typically are elicited after product implementation?
If not, can you tell me by giving another example how the requirements typically are elicited after product implementation?

Guiding questions:
Who is involved in the elicitation of requirements after product implementation?
How much end-users are involved in the elicitation of requirements after product implementation?
What techniques are used to elicited requirements after product implementation?

General questions:

I will now be asking general questions about the process of Requirements Engineering.

How are the people that are involved motivated throughout the process in this method?
How do you ensure the quality of the requirements?
How do you typically measure the success of the product?
How is the process transformed over time?
What challenges do you encounter in your process?
What are the effects of this method of Requirements Engineering?
What software do you use to support this Requirements Engineering process?
Question about crowd-based RE:
I am going to explain a concept which I will ask you a question about. Crowd-based Requirements Engineering is an addition to the traditional Requirements Engineering. The main activities of Requirements Engineering are the same, but the people used in the process of gaining requirements are different. In crowd-based Requirements Engineering developers use a crowd which is a large group of current or potential users. The crowd is continuously involved in the process and after the implementation of the product. The crowd provides user feedback such as software problems, improvement suggestions or new-product ideas. Monitoring systems can be implemented to collect context and usage data of the crowd to better understand the feedback. This feedback can lead to new requirements for the product. Developers want crowd-members to have a genuine interest in the outcome of the product so that their participation is the most effective. Therefore, different approaches are used to motivate the crowd.

Can you estimate how crowd-based your Requirements Engineering process is?
Why do you think that you are (not) using crowd-based Requirements Engineering?
Would you consider doing crowd-based Requirements Engineering? Why (not)?

Closing questions:
Do you have anything you want to elaborate?
Do you have any further questions or comments?
If there are any uncertainties, can I contact you?