A Crowdsourcing Technique for the Requirements Elicitation from Online Reviews

Martijn van Vliet (4171934) (m.vanvliet@uu.nl)
Department of Information and Computing Sciences, Utrecht University
Utrecht, The Netherlands

August 1, 2019

A thesis submitted in partial fulfillment of the requirements for the degree of:
Master of Business Informatics.

First supervisor: Dr. F. Dalpiaz
Second supervisor: Dr. I. Lykourentzou
External supervisor: Eduard Groen

Contents

1 Introduction

1.1 Problem Description

Substantial amounts of user feedback is available from different sources, with app stores and Twitter being the prevalent ones. Making use of user reviews to mine valuable information for improving applications is critical for retaining existing users and attracting new users (Lu & Liang, 2017). As the amount of user feedback grows, it becomes more difficult for humans to manually put in the effort required to assess all user reviews (Groen, Schowalter, Kopczynska, Polst, & Alvani, 2018). This is less than ideal, as it is plausible that valuable feedback will be left untouched or unidentified, which lead to sub-optimal features and missed opportunities.

Different techniques based on natural language processing (NLP) exist that help processing the volume of provided data. These techniques are often the optimal choice from the efficiency perspective, as they are able to process large volumes of data automatically. The results however are often superficial and non-refined, as most techniques employ rudimentary classification techniques (Williams & Mahmoud, 2017). NLP approaches may perform well for simple tasks such as categorizing reviews as informative from a RE perspective versus not informative reviews, but may fail to make finer distinctions such as feature versus bug, or privacy versus security requirements (Dhinakaran, Pulle, Ajmeri, & Murukannaiah, 2018).

Additionally, research on eliciting requirements from a large number of online reviews using these automated means often have a focus on functional aspects (Groen, Kopczyńska, Hauer, Krafft, & Doerr, 2017). This leaves feedback regarding the quality aspects of a software application untouched, although quality requirements are vital for its success (Groen, Kopczyńska, et al., 2017). From the perspective of RE, non-functional requirements are also very much important, as they often represent quality aspects of a system. Studies have indicated that user feedback in online reviews can provide relevant information on app quality, but only on aspects by which users are directly affected (Groen, Kopczyńska, et al., 2017). Therefore, user feedback is mostly focused on usability, performance, efficiency and security. This creates the case for the importance of other quality aspects in this type of requirements elicitation techniques. With the current landscape of NLP techniques, it is assumed that purely automated techniques fall short in this aspect (Groen, Kopczyńska, et al., 2017).

Due to the lacking effectiveness of these automated techniques, alternative processing techniques have been studied under the name of CrowdRE. For instance, techniques that aim to manually extract requirements from user generated feedback also exist (Groen et al., 2018). Manual extraction is possible and results are generally more fine-grained and meaningful, but is also very time-intensive to the point that it often becomes unfeasible due to inefficiency. Furthermore, the quality of the elicited requirements is highly dependent on the knowledge and skills of the human taggers. It is therefore possible that a lot of time and effort is spent on the processing of the data, but that the results will be unsatisfactory for stakeholders. In other words, it is possible that all the spent effort results in a poor return of investment.

For both manual and automated techniques, challenges arise due to the nature of the data. The scale of the data from these kinds of sources, its unique format, diverse nature and high percentage of irrelevant information and/or spam make it harder to properly elicit the relevant information, for both automated and manual techniques (Williams & Mahmoud, 2017). Therefore, both types of techniques require a proper amount of training in the form of adequate training datasets or knowledgeable human annotators in order to become useful.

In this thesis, we surmise a crowdsourcing approach for the analysis of these large datasets as a possible solution for these kinds of problems. Diluting the required workload of human tagging and dividing the required effort by using a crowd, may make it more feasible for organizations to gather usable requirements from this type of user feedback. Additionally, the crowd could possibly also be used to create adequate training datasets in order to improve the results from existing NLP techniques. Either of these approaches are in line with emerging themes in RE which focus on exploiting both artificial intelligence and human intelligence for a complementary approach for the elicitation of requirements (Dhinakaran et al., 2018).

1.2 Approach

The aim of this thesis will be to construct a training method that will help train a crowd (size unspecified for now) to become more knowledgeable about requirements so that they become better at recognizing quality aspects of a system. With this training method, it should become possible to more easily create a crowd that has sufficient knowledge about requirements. This crowd could then potentially be used for either manual extraction of requirements or for tagging purposes that allow the creation of training datasets for automated techniques. Either way, this requires a lightweight method that instructs the crowd to adequately distinguish and categorize different types of requirements.

A visualization of the process that will realize this project is shown on the next page.

1.2.1 Overview of Approach

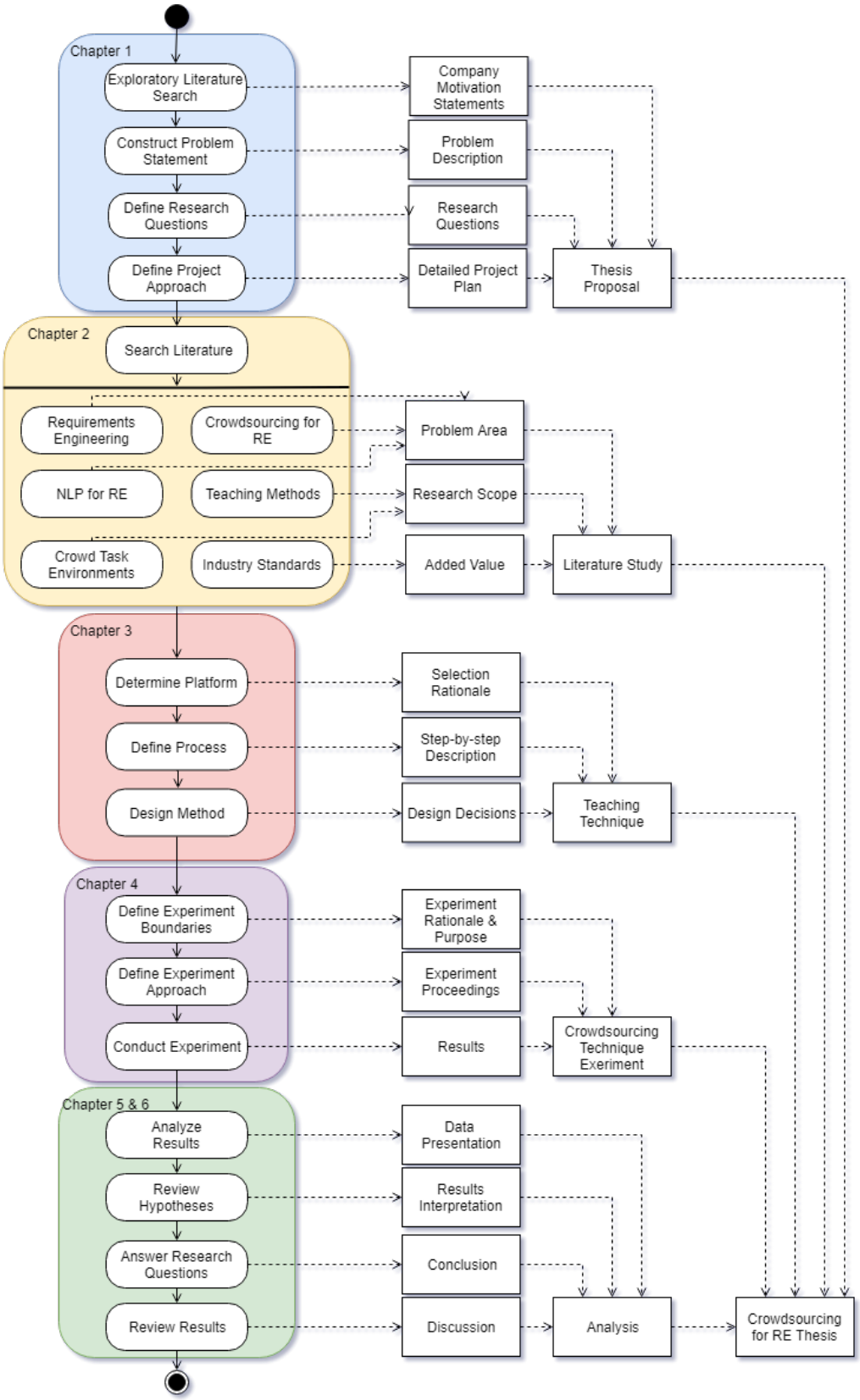


Figure 1: A high level overview of the Thesis Project.

1.2.2 Literature Study

The structured literature study should focus mainly on the following topics:

- What kind of requirements are there and what are their characteristics?
- What is the quality of existing NLP techniques in the context of RE?
- Identification of challenges of these kinds of elicitation techniques and their nature?
- General teaching and training methods/approaches?
- What environments exist that could facilitate crowdsourcing tasks?
- What is the current state of crowdsourcing (both for similar purposes as for RE or other)?
- What are industry trends/standards on how organization handle the feedback from sources such as app stores?

1.3 Research Questions

Based on all of the previous information, the following research questions were derived:

Main RQ: ***How can a method be constructed that facilitates the elicitation of requirements by using a sizeable crowd of non-experts?***

The main research question will be addressed by answering the following sub questions.

1. What design characteristics have to be involved for the design of a crowdsourcing technique?
2. Are crowd workers able to distinguish relevant user reviews that are useful for developers from irrelevant reviews?
3. Do crowd workers have the ability to classify user reviews into requirements categories?
4. Will crowdsourcing RE tasks be a feasible solution for dealing with challenges accompanying user generated feedback from online sources?

In addition to a comprehensive design approach, it is likely that real-world test will have to be performed to answer the questions regarding the feasibility of a crowdsourcing method. The exact shape of the test will be determined at a later stage. However, the focus of the test and its goals can already be derived from the problem statement.

The hypotheses that the test will assess are:

H1. Crowd workers have the ability to recognize the differences between useful and useless reviews.

H2. Crowd workers are able to correctly categorize user reviews in different requirement categories.

H3. A crowdsourcing technique for the elicitation of requirements from online reviews is a feasible and cost-effective approach.

The hypotheses described above aim to test the effectiveness and usefulness of the to be developed elicitation technique, with effectiveness referring to the capabilities of the crowd. Usefulness in this case is defined as a degree of quality of the outcome, catered to the desires of involved stakeholders. For instance, should the technique be tested and applied to a real case, the main goal is to gain output that has more value for stakeholders. In practice, this could result in either a higher number of usable requirements from a set of reviews or the same number of usable requirements acquired with less effort compared to regular practices.

1.4 Contributions to Science

The conduction of this project provides interesting insights for academia. By means of this project, insights on the performance and effectiveness of training methods and how usable they are in the context of RE will be provided. Additionally, a detailed description is provided that documents the construction of the training method. Characteristics of crowdsourcing techniques are researched for the construction of a framework to guide the decision making process in the construction phase. This enables the ability to review the design decisions to reflect on what parts worked correctly or worked less than ideal, and whether the assumptions made in these circumstances were correct.

Apart from the training method, the project provides better insight of what the effects of crowdsourcing are on the quality of the elicited requirements, considering both functional and non-functional requirements. Existing techniques are lacking in the identification of quality requirements, so an adequate crowdsourcing solution can be considered a valid contribution to the academic world.

Finally, the insights gained from the project itself with an accompanying review at the end can be used to further refine the process of using crowdsourcing for RE.

1.5 Contributions to Business

While the method was not tested and validated by using one single real-life case for an existing business, the project did result in a couple of contributions. First and foremost, this project analyzed an existing pool of unstructured data of user feedback. Out of this analysis, usable information was elicited for the products involved in the datasets. Therefore, the project has demonstrated that it is possible to yield useful information from similar sets of data. Due to this, it allows organizations to explore crowdsourcing as a way to process their user feedback and to elicit usable requirements. The method resulting from this project might in turn be used by other businesses themselves, which could improve the quality of their gathered requirements in a way that requires less effort and man-hours.

2 Literature Study

The literature study this thesis covers different topics, each contributing to the overarching research questions. Each upcoming section explores a dedicated topic, so that knowledge can be gathered that will contribute to reaching the goal of this thesis: which is the creation and testing of a crowdsourcing technique for requirements elicitation and analysis.

In section 2.1, we focus on requirements engineering in the general sense to establish the baseline of this thesis project, and explore challenges and opportunities. Afterwards in section 2.2 and 2.3 respectively, the role of both crowdsourcing and Natural Language Processing (NLP) in RE will be explored by analyzing current techniques to see how they fit into the previously identified challenges and opportunities. By including the two topics of crowdsourcing and NLP for RE, remaining challenges and opportunities may be narrowed down to facilitate a possible alignment with the focus of this thesis project.

Additionally, an insight into industry standards and practices regarding RE practices will be provided through means of an interview in section 2.4. The goal of exploring industry standards is to gain insights in established methods that already deal with user generated feedback. These insights are gained substantiated with existing literature and where possible, to fortify the insights given by the representatives from companies included in the interview.

Lastly in section 2.5, the focus is shifted towards the actual construction of a crowdsourcing technique. Aspects of general teaching will be explored as well, in order to discover the best approach on how to properly teach a crowd. Ideally, the crowd will be taught in a way that they become more knowledgeable about requirements so that they can utilize their newly acquired knowledge as effectively as possible.

No rigid protocol for the literature review process was constructed beforehand, but a more flexible approach was applied. The research started from a couple of pointers towards established, well-known and highly referenced literature and articles related to the discussed topics found by using Google Scholar, and relied on snowballing from there.

2.1 Requirements Engineering

For any kind of developed software product, it is important that it matches and fulfills the desires of its users. The primary measure for the success of a software system is often expressed as the degree to which it meets its intended purpose (Nuseibeh & Easterbrook, 2000). Furthermore, poorly handled requirements and the inability to incorporate user needs into a products are often accredited as a major cause of failed software projects (Chemuturi, 2012).

It is therefore important for organizations that rely on the success of their software product(s) to manage their requirements in a proper way. The discipline that manages these desires is defined as Requirements Engineering (RE) and concerns the elicitation, analysis, specification and validation of these desires (Bourque, Fairley, et al., 2014). The importance of high-quality RE and its importance for the development of software has been recognized for quite a while now (Nuseibeh & Easterbrook, 2000). This chapter concerns the fundamentals of the RE discipline and its evolution over time to gain a better understanding of its practices and their challenges. Understanding current challenges and how they changed over time will aid the exploration and identification of opportunities that will be useful for later stages in this project.

One of the classic definitions of RE is provided by Zave (1997): RE is *"...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."* This definition encompasses the core responsibilities of the RE discipline, specifying and realizing real-world goals and tracking how they change and evolve over time.

Traditionally, RE focused on discovering what developers should build before actual development of the software product commenced (Paetsch, Eberlein, & Maurer, 2003). This resulted in the

following process shown in Figure 2.

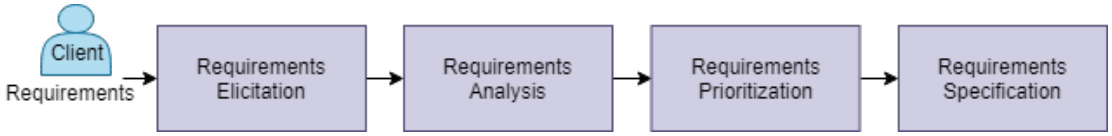


Figure 2: The traditional RE process as stated by Paetsch et al. (2003).

According to the traditional RE approach, the RE process starts linearly from **eliciting requirements** from a client even before software development commences (Paetsch et al., 2003). The RE discipline offers a plethora of different techniques that can be used by software development organizations to manage requirements to further shape their product during its life cycle. These requirements can be elicited from different sources, such as business concerns and corporate goals, all kinds of stakeholders and the domain itself where the software is supposed to operate in (Bourque et al., 2014). Several traditional elicitation techniques proposed by (Preece, Rogers, & Sharp, 2015) are:

1. **Interviews:** Meeting stakeholders to explore issues in a structured, semi-structured or unstructured manner.
2. **Scenarios:** Creation of informal stories about user tasks and activities to express work situations to aid the conceptual design.
3. **Focus groups:** The creation of workshops where a set of users brainstorm about their needs of a system and discuss them.
4. **Observation:** Observing how users interact with a system in order to better understand the users' context, tasks and goals.
5. **Prototypes:** A more comprehensive method where users interact with an early version of the system to gain feedback early in the development process.

All techniques explained above have in common that they are human-centered, illustrating the human dynamic within RE practices as they are all dependent on the involvement of human input. The context in which RE takes place is usually a system that requires this human input, with the aim to solve human owned problems (Nuseibeh & Easterbrook, 2000). Most of these techniques are not quick or easy to conduct, either because they require a lot of preparation or because they require face-to-face contact with a set of users.

When the requirements have been elicited from a client, they need to be **analyzed** so that they can be used for development purposes. They are checked for necessity to determine the actual need of a requirement, consistency so that they are not contradictory, completeness to make sure that no function or constraint is missing and feasibility to check how much effort implementation may take (Paetsch et al., 2003). It is possible that different stakeholders (i.e. users or developers) have contradicting ideas about the future of the software product, which will create conflicts between different requirements.

If the initial analysis of the requirements leaves a conflicting set of requirements or when organizational resources are too limited, further conflicts can be solved by applying **prioritization** techniques. Resolving conflicting requirements can be achieved by reviewing them on other aspects, and are different for each specific case or even for different iterations of the same software product. Some of these reviewing aspects for further refining the prioritization of requirements are illustrated below (Paetsch et al., 2003; Bourque et al., 2014).

1. **Necessity:** Based on the amount of current critical bugs or based on legal reasons or similar constraints.
2. **Resources:** Comparing the amount of money and time a feature requires to be implemented with the available budget or current schedule of the organization.
3. **Customer importance:** The degree of satisfaction a client or a user would reach when a specific feature is implemented.
4. **Business Priorities:** Business objectives set by higher management that determine the strategy of the software development organization.

Determining how important certain requirements are should be done by both the users and the developer. The customer should mark features providing the greatest benefit to users with the highest priority. Developers point out the technical risks, costs, or difficulties so that they together figure out the optimal solution (Paetsch et al., 2003). Proper RE practices are important as development organizations have limited resources, so correct prioritization becomes critical to not waste resources on undesired or unnecessary features.

In addition to the prioritization of stakeholder desires, **specification** ensures that they can be used and managed during development phases. In software engineering, specification typically refers to the production of a document that can be systematically reviewed, evaluated, and approved (Bourque et al., 2014). Requirements specification considers three dimensions, where the specification refers to the degree of requirements understanding at a given time (Pohl, 1994). Moreover, the second dimension proposes that the representation of a requirements can be either in informal or formal language. The model shown in Figure 3 illustrates that the specification process transforms opaque and informal requirements to the desired output of the third dimension, which is the agreement on a common view of the specification of a formally defined requirement (Pohl, 1994).

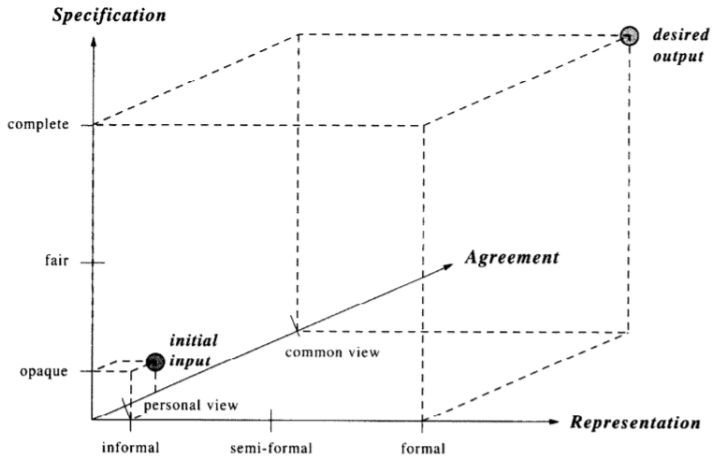


Figure 3: The three dimensions of RE defined by Pohl (1994).

Specification aims to create a common understanding of terminology due to challenges introduced by the human factor involved in RE practices. The fact that different people may refer to identical system features in different ways complicates both the entire RE process and the development process, as it may introduce misunderstandings between developers (Dalpiaz, van der Schalk, & Lucassen, 2018). This problem directly scales with the amount of people related to the RE practices described above, so adequate requirements specification becomes more important for large organizations or very complex systems. Adequate specification allows for the requirements to be validated to certify that the requirements are a correctly fitting description of the system (Paetsch et al., 2003).

2.1.1 Requirements Types

A requirement in itself is defined as *...a statement about an intended product that specifies what it should do or how it should perform* (Preece et al., 2015). This definition distinguishes between different types of requirements, either focused on concretely defining what a system should do or setting standards as to how the system should perform quality wise. Therefore an important distinction has to be made between two different kinds of requirements.

*"**Functional Requirements (FR)** describe the functions that the software is to execute; for example, formatting some text or modulating a signal" (Bourque et al., 2014).*

*"**Nonfunctional Requirements (NFR)** refer to quality aspects of a system and place constraints*

on the development possibilities (Preece et al., 2015).

The nature of the two different types of defined requirements have implications for their use in the development process. Even though it is recognized that the utility of a system is defined by both FRs and NFRs, historically there has been a lop-sided emphasis on FRs even though they are not effective without the necessary characteristics offered by NFRs (Chung & do Prado Leite, 2009). This emphasis stems from the fact that NFRs express properties of a system as a whole, which makes it harder to attribute them to individual components (Nuseibeh & Easterbrook, 2000). Non-functional requirements represent emergent properties of software, which are requirements that cannot be addressed by a single component but which depend on how an entire system and all its components work together (Bourque et al., 2014). This in turn makes them more difficult to analyze which inhibits the creation of an adequate judgment about the overall quality of a system. While the rationale behind this emphasis on FRs may be understandable, as they are more easily testable and have a focus on directly satisfying users, it does have repercussions. NFRs are indicated to be critical for the success of a software product, and the general lack of focus on NFRs and their quality aspects are indicated as one of the factors why software projects fail (Glinz, 2007).

2.1.2 Challenges & Opportunities

While the definition of RE proposed by Zave (1997) still encompasses the core responsibilities of RE, a lot has changed over the years. Firstly, the nature and focus of RE practices have shifted from strictly software related to become more oriented towards entire systems and their dynamics within an environment, further sophisticating the software engineering process (Nuseibeh & Easterbrook, 2000).

Secondly, the rapidly changing business environment in which organization operate in today has challenged the traditional RE way of working (Ramesh, Cao, & Baskerville, 2010). Due to the fact that requirements are volatile, and may change many times as the development process progresses (Bourque et al., 2014), it required business to adapt more agile ways of working in order to keep up with these rapid changes (Fernández & Wagner, 2015). However, these new ways of working relied on short development cycles that require a continuous flow of user feedback parallel and at similar speed to the development cycle. Early research already suggested that the the quick iterations required for agile development did not match with traditional, mostly slower RE practices which makes them unfeasible to be used in an agile environments (Inayat, Salim, Marczak, Daneva, & Shamshirband, 2015)(Paetsch et al., 2003).

While these two developments were in the right direction and contributed to a higher quality of produced software in a general sense, it further complicated the nature of RE practices. To keep up with these developments, RE practices are required to address more complicated aspects but also should be executable in less time to be usable in quick development cycles. This growing complexity and increased time pressure introduce new challenges.

While the software engineering industry recognizes the benefits of proper RE practices, it is still common that software products fail, which can partly be attributed to improper handling of these challenges (Charette, 2005). Currently, the RE industry is still struggling in reaching the desired level of high quality and creating proper standards for it (Fernández & Wagner, 2015).

This lack of high-quality standards and the resulting inaccurate RE practices are linked to 47% of projects that fail to meet their initial goals and business objectives (Smith, 2014). Further survey studies discovered more exact causes for complications from poor RE management practices in development organizations (Fernández & Wagner, 2015). The top three requirements related causes for complications mentioned by development organizations are shown in Table 1.

Nr.	Rank	Cause for Complications	% Mentioned
1.	1.	Incomplete and/or hidden requirements.	48%
2.	3.	Moving targets (changing goals, business processes and/or requirements).	33%
3.	4.	Underspecified requirements that are too abstract.	33%

Table 1: Requirements related complications mentioned by developers.

While these complications may have varying origins, they have in common that they are all detrimental to the success of the software product. This indicates that used requirements management practices applied in the real world do not fully possess the capability to prevent these complications. The origins of these problems however cannot be linked to a single cause, as they differ between specific cases.

For traditional RE practices, the causes for these challenges are often linked to the fact that it is challenging to engage and involve a large number of users in requirements engineering practices (Groen, Seyff, et al., 2017). Additionally, researchers argued that organizations often employ a too narrow concept of a "user" and rely on a recruited set of users that the development organization itself considers to be representative of the entire user base (Snijders, Dalpiaz, Hosseini, Shahri, & Ali, 2014). Furthermore, agile development methods often assume an ideal customer representative, which is expected to answer all developer questions correctly and is able to make the right decisions (Paetsch et al., 2003). In practice, the question remains how representative this image of an ideal user is compared to the average user of a software product. The combination of these arguments can be interpreted as a call towards the involvement of a more diverse and skillful group of users that are both familiar with the case specific systems and RE practices during software development stages.

Using online reviews may allow companies to have access to a pool of user feedback created by a diverse group of actual users and therefore will have gained access to more representative feedback. Even with the troubling nature (Williams & Mahmoud, 2017) of the data from online reviews as discussed in section 1.1, the information may prove to be too valuable to pass up on. However, incorporating this huge pool of user feedback into development processes comes with challenges on its own. With the performance of NLP techniques being unsatisfactory in this regard (Groen, Kopczyńska, et al., 2017) and the effectiveness of manual labor being ineffective (Groen et al., 2018), crowdsourcing makes a compelling argument to be a suitable candidate to act as a solution in between these two worlds.

2.1.3 Conclusion

The previous sections showed how the evolution of both the RE discipline as the software engineering industry contributed to higher quality software solutions, but introduced new challenges as well. From the identified challenges in traditional and modern RE practices, arguments can be derived that justify a need for innovative practices that deal with these new challenges. The proposed solution in this thesis project to combine crowdsourcing techniques with available sets of user reviews may turn out to be a suitable approach to deal with the current challenges. The arguments for the utilization of crowdsourcing in RE practices as discussed in this chapter are summarized as follows in Figure 4.

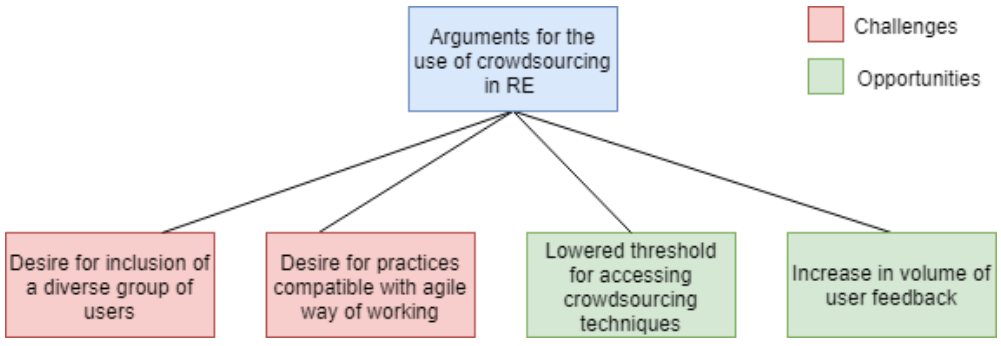


Figure 4: Summary of the arguments for the utilization of crowdsourcing in RE practices.

As shown in Figure 4, the challenges in traditional and current RE practices create an opportunity for the utilization of crowdsourcing techniques in RE practices due to the following reasons:

1. **Desire for inclusion of a diverse group of users:** The recognition that feedback provided by a larger and more diverse group of users is likely to give more useful feedback.

2. **Increase in volume of user feedback:** Compared to the scarcity of user feedback that traditional RE challenges mentioned, the current availability of user feedback in the form of online reviews offers new possibilities.
3. **Desire for practices compatible with agile way of working:** With current NLP practices deemed unsatisfactory and traditional methods recognized too slow and work intensive, an alternative method that fits into the current agile working environment becomes more attractive.
4. **Lower threshold for accessing crowdsourcing techniques:** The increased availability of platforms where crowd-sourced work can be requested and fulfilled lowers the threshold for using crowdsourcing methods in practice.

In the upcoming section crowdsourcing in itself will be studied more thoroughly to determine in what way it can specifically contribute to the RE field.

2.2 Crowdsourcing

The problems and challenges in the RE discipline described in the previous section have shown the opportunities for the utilization of crowdsourcing and to indicate possible contributions to RE practices. The usefulness of collective intelligence has been recognized from early on (Lévy & Bononno, 1997), being one of the reasons for the emergence of the practice of crowdsourcing (Howe, 2006). Crowdsourcing is defined as: *"...the act of taking a task traditionally performed by a designated agent (such as an employee or a contractor) and outsourcing it by making an open call to an undefined but large group of people"* (Howe, 2006).

This section studies the benefits and challenges of crowdsourcing. It discusses the main methods and practices of existing crowdsourcing methods to gain a better understanding of the dynamics between the crowd and the tasks to be performed. It explores the different ways crowdsourcing can be utilized to facilitate the construction of an adequate method for RE practices and to prevent common pitfalls.

2.2.1 Types of Crowdsourcing

Bigham, Bernstein, and Adar (2015) identified three different types of collective intelligence for the application in the field of human-computer interaction (HCI). This section explores the characteristics, benefits and challenges of these three types of collective intelligence applied in the context of crowdsourcing.

1. **Directed Crowdsourcing:** A person or an algorithm directs workers to pursue a specific goal. In this form of crowdsourcing, crowd workers complete tasks that are directly asked to be completed. Challenges may arise when the interest of the crowd and the requester are not aligned, as the workers may need additional incentives to perform their tasks. Monetary rewards are one option, but often result only in an increase in quantity and not necessarily the quality of the performed work (Mason & Watts, 2010). Therefore, directed crowdsourcing is required to have some form of quality control implemented to create useful results. Additionally, this form of crowdsourcing is often argued to be most suited for tasks that workers are expected to perform well with little to no training. Further complications for the quality of work can be expected when the crowd is subjected to tasks that they deem to be too difficult.

2. **Collaborative Crowdsourcing** Does not rely on directed forms of labour, but relies on the intrinsic motivation of the workers themselves. This form of crowdsourcing is called collaborative crowdsourcing and has shown to be the driving force behind projects such as the creation of Wikipedia. This form of crowdsourcing requires workers to have some sort of emotional investment or connection to the work to be done to ensure effective participation. Depending on each case, it can be challenging to arise enough intrinsic motivation for users to collaborate in similar projects for businesses when this connection or investment is not present. This form of crowdsourcing is also susceptible to internal conflicts, due to lack of a clear leadership structure. This is not solely an impairment, as it could also incite discussion and debate among the crowd which could lead to improved insights or new aspects. However, this depends completely on the knowledge level of the involved crowd. This form of crowdsourcing especially relies on on knowledge that is already in the head of the workers, which can be observed as one of the main challenges.

3. **Passive Crowdsourcing** The relationship between requester and the crowd can also be indirect. In passive crowdsourcing, the crowd produces useful output as a result of their normal behavior. In other words, the work is a side-effect of what people are doing ordinarily. Instead of directing the efforts of the crowd, the requester is passively monitoring the performance of the crowd. Dedicated infrastructure is required to prevent interference and to deduct usable and valid results from way of working. Preventing interference is crucial for this type of crowdsourcing, as the quality of results can only be assured when the system is properly observing ordinary behavior. This type of crowdsourcing does not require direct work requests to be made towards the crowd, as results are generated passively.

Regardless of the type of crowdsourcing, the definition of crowdsourcing by Howe (2006) implies that the tasks to be performed contribute to a common goal. Moreover, Howe (2006) states that

an unspecified, but large group of users can be involved in crowdsourcing activities. Having an unspecified group of people defined is not without a reason, as it gives an indication about the possible flexibility for crowdsourcing activities. To give an illustration about this flexibility, some of the possible crowdsourcing activities defined by Parvanta (2013) and Estelles (2015) are shown below:

1. **Crowd funding:** Raising money for (mostly) new and innovative initiatives by individuals or organizations in exchange for a reward. This form of crowdsourcing increases the chance for riskier initiatives to acquire their desired funding, as the investment risk is spread among the entirety of the crowd.
2. **Crowd labour:** Recruiting individuals to create a crowd that performs specific tasks, ranging from very small to large complex tasks such as image classification or text translation.
3. **Crowd research:** The gathering of insights from intended audiences, mostly used for marketing research or public opinion research purposes by using surveys.
4. **Creative crowdsourcing/Crowdcasting:** Motivating a crowd to solve creative problems such as design contests for product development purposes, such as logos for rebranding. Usually presented in the form of contests where a problem or tasks is proposed, promising rewards for the quickest or the best solution.
5. **Crowd collaboration:** Gathering a crowd that is directed to work together, either for brainstorming or supporting purposes. Here, the crowd is directed to come up with different solutions for a case presented by a director by working together, or is directed to collectively provide for instance customer support to fellow customers.
6. **Crowd content:** Directing a crowd to use their labor and knowledge to create or find solutions to a problem in a non-competitive way. Usually applied where the crowd is supposed to create or search usable input or can analyze and provide feedback from large pieces of work.

In addition to the flexible possibilities of the harnessed power of a crowd, additional benefits for the utilization of crowdsourcing exist. Firstly, crowdsourcing offers several ways to gather input very quickly and cheaply (Parvanta et al., 2013), opening new opportunities for organizations. Secondly, it appears to broaden participation as it opens up the possibility for additional people to participate in problem solving, which is linked to better outcome and improved results (Jeppesen & Lakhani, 2010).

Nevertheless, these benefits can only be reaped when the crowdsourcing activities and methods are performed and applied in a correct way. Inadequate practices negatively affect the quality of the results generated, which appears to be a traversing trend between all crowdsourcing models crowd types and sizes showing challenges in quality assurance (Adepetu, Ahmed, Al Abd, Al Zaabi, & Svetinovic, 2012). Combined with the idea that crowdsourcing has traditionally shown to be most effective for problems that required little expertise (Bigham et al., 2015), assuring an adequate level of quality for more complex tasks remains a challenge. It is therefore crucial to further understand the dynamics between possible tasks to be performed and the crowd in a crowdsourcing setting, to ensure the construction of a correct method compatible with the goal of this thesis project.

2.2.2 Task Characteristics

The amount of different crowdsourcing practices illustrate a wealth of possibilities for their utilization. However, while the exact application may vary, they do have in common that they work towards a common goal. As the definition of crowdsourcing by Howe (2006) indicates, crowdsourcing works towards the completion of a task or a series of tasks. The nature of the tasks to be completed therefore shape the specifics of the crowdsourcing process.

Schenk and Guittard (2011) define two dimensions that shape these specifics: the nature of the process itself and the type of tasks. They state that the goal of a crowdsourcing process is to either pool complementary input of a crowd together or to gain access to problem solving capabilities. Pooling complementary input of a crowd is useful for building an information base required for further decision making. Gaining access to problem solving capabilities on the other hand is mostly relevant when a very specific goal or task has to be fulfilled.

Complimentary, Schenk and Guittard (2011) also define three categories that allow the classification of the different types of tasks found in crowdsourcing practices. A summary of these categories and their characteristics is shown in Table 2.

Type	Expected Contributions	Knowledge Required	Benefits	Downsides	Required Incentives
Simple Tasks	Data or Information	- Low - Mostly Small and Routine	Low Costs	- Requires a large scale to be effective.	Low
Complex Tasks	Problem Solving	- High - Expertise Required	Availability of a wide range of competences.	- No guarantee to find correct solution. - Costly to incentivize crowd. - High dependency on crowd capabilities.	High
Creative Tasks	Creation of Content	- High - Creativity Required	Diversity and Novelty	- No control over output. - High dependency on crowd capabilities.	Variable

Table 2: The three categories of crowdsourcing tasks defined by (Schenk & Guittard, 2011).

The three categories show that possible negative implications for the results directly scale with the level of complexity of the tasks. Complex tasks require a high level of organization and processing due to the challenge of dividing problems, assigning the sub-problems to individuals and combining the solutions (H. Zhang, Horvitz, Miller, & Parkes, 2011). Simple micro-task workflows are shown to be the dominant crowdsourcing structure, but only enable goals that are so simple and modular that their path can be entirely pre-defined (Valentine et al., 2017). Having multiple people solve the same simple tasks and aggregating the results in a more comprehensive solution is a way to improve the quality of the results, and has been demonstrated successfully in the past (H. Zhang et al., 2011). Therefore, attempting to split the entirety of the task to be done into as many easier sub-tasks as possible appears to be the most desired implementation for the processing of more complex tasks.

2.2.3 Crowd characteristics

According to the definition of crowdsourcing by Howe (2006), the work to be done is outsourced through an open call to which basically anyone can respond. The workers who respond to an open call are unknown to the organization that needs the work done. This introduces risks for the organization, as complex tasks require a certain level of knowledge and skills to be completed successfully (Schenk & Guittard, 2011). Therefore, for companies to extract value from crowdsourcing practices, they must match the right crowd to their specific organizational needs (Bogers, Afuah, & Bastian, 2010).

The potential of the crowd as a source of collective intelligence has been recognized (Wexler, 2011), but requires the co-design of technical infrastructure and human interaction to utilize (Bigham et al., 2015). As both these factors can take many forms, the challenge lies with finding the right dynamic between the technical infrastructure, the interaction process and the characteristics of the crowd itself to make optimal use of their capabilities.

The capabilities of a crowd itself rely on a wide variety of different skills, knowledge, cognitive strategies, experiences and problem solving approaches (Mumford, 2003). Research conducted by Erickson, Petrick, and Trauth (2012) resulted in the division of crowd characteristics into three themes to define the overall capabilities of a crowd.

1. **Crowd Knowledge:** The capability of a crowd to match an organizational need and complete their tasks based on their knowledge. This type of knowledge is divided into five different types: general knowledge, situational knowledge, product/service knowledge, specialized knowledge and domain expertise.
2. **Crowd Value:** The crowd values defines how much value a crowd could potentially bring to the task that they will perform. Crowd value is based on the diversity of a crowd, their total distributed knowledge that they bring to the table and lastly, their sheer numbers.
3. **Crowd Location:** Defines whether the crowd is made up of external or internal sources from the perspective of an organization. The source of the crowd is important as external

sources tend to improve the diversity and size of the crowd, but also introduce potential risks and noise compared to internal sources due to lack of control.

Erickson et al. (2012) also linked their three themes with organizational needs to indicate the desired type of crowd knowledge, value and location to achieve optimal benefits. They constructed their framework based on grounded theory principles and information retrieved from interviews with practitioners, reviewed literature and exploratory case studies. The framework constructed by Erickson et al. (2012) is shown below in Figure 5.

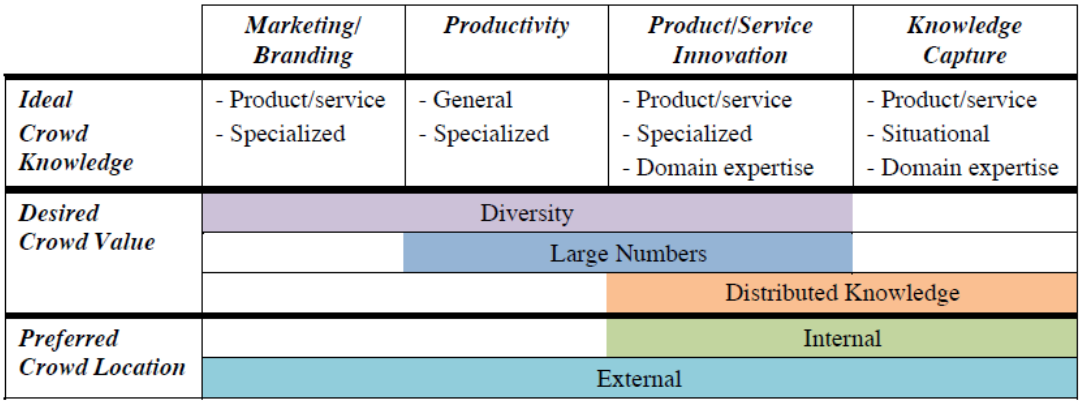


Figure 5: Crowd characteristics matched with organizational needs (Erickson et al., 2012).

Through their model, Erickson et al. (2012) show that different types of organizational needs require different focuses on each of the three themes to achieve optimal results. For instance, crowdsourcing for marketing purposes benefit the most from a diverse crowd that has both specific knowledge about a certain product or service, as well as some specialized understanding about marketing. The most interesting organization need in the model is the product/service innovation need, as it takes advantage of almost all of the involved aspects. For innovation, a large, diverse crowd with a high level of distributed knowledge all contribute significantly to the increased chance of generating new revolutionary ideas (Erickson et al., 2012).

2.2.4 Crowdsourcing in RE

Crowdsourcing practices have already entered the world of software engineering (LaToza & van der Hoek, 2016). This is also the case when looking explicitly at RE practices, as crowdsourcing has been identified as a scalable and inexpensive way to involve users in RE practices (Snijders et al., 2014). Existing literature calls for the use of crowdsourcing to deal with current challenges in RE (Hosseini, Phalp, Taylor, & Ali, 2014). The involvement of user perspective into development practices has shown to play an important role with keeping up with the rapidly changing environments that current software systems operate in (Ali et al., 2011).

While the role of the user has always been present in traditional RE practices, the focus on user participation has been mostly on the elicitation process of requirements. The emergence of different forms of crowdsourcing have opened up the ability for better crowd involvement during not only the elicitation process, but for analysis and prioritization as well (Snijders et al., 2014). This is illustrated in Figure 6.

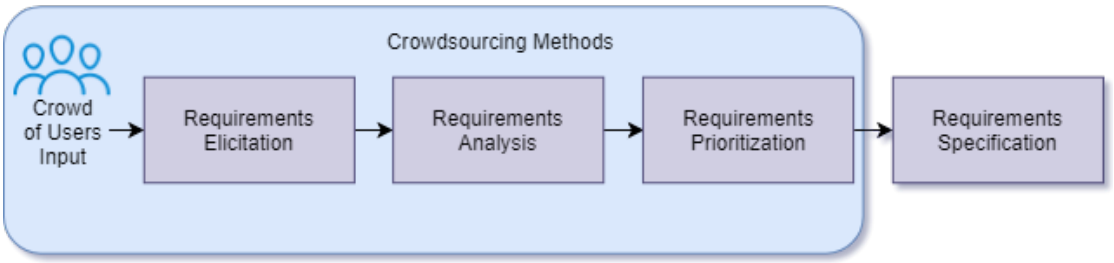


Figure 6: Envisioned crowdsourcing methods contributing to traditional RE Practices (Snijders et al., 2014).

Applying crowdsourcing techniques in the development process enables feedback-based requirements engineering, indicating that users can directly contribute towards the understanding of the requirements during the development of the next version of the software (Hosseini et al., 2014). In a sense, the crowd has already been passively crowdsourcing the elicitation part by leaving online reviews about their subjective performance of a system. While this explicit type of user feedback is recommended to be taken into consideration during development to keep up with the ever changing system environment and context, it is unclear how practitioners can integrate this continuous flow of feedback into their practices optimally (Maalej, Nayebi, Johann, & Ruhe, 2016). Integrating this large amount of feedback into development processes is beneficial, as it has the potential to increase the quality and comprehensiveness and even the economic feasibility of requirements elicitation (Hosseini et al., 2014). However, this practice also introduces new challenges. The sheer volume of online reviews require a high workload to process this type of feedback manually, making it unfeasible due to cost and time constraints (Groen, Seyff, et al., 2017). These relatively intensive but slow practices threaten the validity of the results in dynamic environments, as they are likely to be valid only temporarily (Hosseini et al., 2014).

Utilizing crowdsourcing for not only the elicitation but also the analysis and prioritization of requirements may offer a solution for these challenges. The initial filtering of low-quality results from the set of online reviews can be requested from the crowd and be presented as micro or simple tasks. Further analysis and classification of the results can be presented as more complex to a more specialized crowd to support the requirements engineers and reduce their workload, potentially shorten the time to market (LaToza, Towne, Van Der Hoek, & Herbsleb, 2013). When implemented correctly, utilizing these forms of crowdsourcing is cheaper than complete outsourcing or having own dedicated employees (Adepetu et al., 2012). Lastly, utilizing the potential of a large enough crowd allows for the quick processing of huge volume of user feedback, making it compatible with agile development methods (Mao, Capra, Harman, & Jia, 2015).

To reap these benefits, there has to be dealt with the challenges that these crowdsourcing methods introduce as well. Especially for the more complex tasks, a certain level of knowledge and expertise is required from the crowd to assure an adequate level of quality that is plaguing crowdsourcing practices in general (Adepetu et al., 2012). Ideally, the crowd also has some intrinsic motivation and some kind of emotional investment and knowledge in/about the subject or product of the online reviews. Therefore, a mismatch between the crowd and the organizational needs of a development organizations has to be prevented. Furthermore, the practices should be implemented in a way that the effort required to properly aggregate the results and deducting the right conclusions is as low as possible. This includes compensating for the general lack of focus on quality aspects from these kinds of user feedback sources (Groen, Seyff, et al., 2017). Lastly from an organizational perspective, involving the crowd into their RE practices also requires a certain degree of openness from organizations (Snijders et al., 2014). It is plausible that not all kinds of organizations are open to that either due to legal constraints or brand or product protection.

2.2.5 Existing Methods and Techniques for CrowdRE

While the potential of the utilization of crowdsourcing techniques in RE has been acknowledged, relatively few attempts have been made in practice (Hosseini et al., 2014). Nevertheless, concrete methods have been made available that aim to guide software product companies in involving a

crowd of users during RE practices more effectively (Snijders et al., 2014). This section explores some of these methods and describes them from the perspective of the original authors, to learn from their practices and their encountered challenges.

StakeRare

The Stakerare method developed by Finkelstein and Lim (2012) aims to identify and prioritize requirements for large-scale software projects. The authors developed their method triggered by the observation that large-scale software projects introduce new challenges for the management of requirements. They state that large-scale software project often suffer from information overload, inadequate stakeholder input and biased prioritization of requirements. The causes for these complications are linked to scaling issues from existing elicitation techniques, unrepresentative inclusion of stakeholders and existing prioritization techniques being too dependant on individuals.

Figure 7: The StakeRare process by Finkelstein and Lim (2012).

To deal with these challenges, the StakeRare method helps to identify and prioritize requirements using social networks and collaborative filtering. The method builds a social network of the involved stakeholders and asks them to individually prioritize a set of requirements. The stakeholders only review requirements that are relevant to them, based on the characteristics of their individual profile. The method gathers the results for different categories of stakeholders and reaches a prioritization based on their scores and their rated influence in the project.

The StakeRare method was evaluated and applied to real-world large-scale software projects. They compared a list of prioritized requirements resulting from the StakeRare method with a list of requirements derived from conventional methods. The evaluations showed that StakeRare had similar performance in regard to precision and recall of requirements compared to the conventional method, and showed a high correlation with the prioritized results. The method also demonstrated less overall time spent to get a list of prioritized requirements compared to conventional methods and that the involved stakeholders preferred this way of working, as it required less effort.

CrowdREquire

Adepetu et al. (2012) developed a platform for the support of RE practices based on a crowdsourced workforce. The authors focused on multiple areas within the RE discipline and included elicitation, analysis, specification and quality assurance. The platform is developed to offer services in a web-based environment for projects submitted by individuals, corporations or other external bodies. The crowd involved in providing these services would be incentivized by creating an atmosphere of healthy competition among the users, by providing cash rewards and by providing them with a sense of prestige.

The authors approach the development of the platform from four different angles: crowdsourcing, requirements specification, business model and market strategy. Regarding crowdsourcing, they argue that a successful crowdsourcing platform should have a structure to guide the clients and the crowd and to assist the task defining clients and problem solving crowd members. In addition, they state that the crowd should be of a sufficient size and should be motivated to participate by making the platform easy to use. The authors are aware of general quality assurance challenges within the field, and aim to address these by providing training questions to ensure that their workers are qualified. They assume that these tests in combination with a competitive environment naturally ensures that crowd members will perform to the best of their abilities.

The goal of the crowd is viewed as to produce specification of requirements in accordance to a specific template which adheres to industry standards. They do not interfere with further processing or judgments, as they leave the final say on acceptance of the submitted requirement specifications up to the client itself.

The platform is marketed as a distributed problem solving medium that is globally accessible. They offer clients the possibility to submit requirement specification tasks that are supposed to be solved by members of the crowd, but no concrete implementation of this process is defined. CrowdREquire is defined as a conceptual model and no real cost assessments or feasibility analysis were performed. The authors propose a set of experiments that can be conducted to further explore the potential of their model, but no evidence can be found that these experiments have been conducted as of yet.

CRAFT

CRAFT stands for Crowd Annotated Feedback Technique and is developed by Hosseini, Groen, Shahri, and Ali (2017). This technique is developed from the perspective that machine learning and other automatic means do not perform adequately enough for the elicitation of requirements. The authors recognize the potential of user generated feedback on social media and other user platforms, but state that the unique nature of natural language is too challenging to analyze automatically. Feedback gathered from these sources introduce challenges, as the users usually provide their feedback in their own words and expressions, introduce noise and errors and tend to hint at a problem instead of stating it explicitly.

The authors therefore proposed a crowdsourcing technique that allows the crowd to annotate the feedback provided by users. The technique is structured in a way that a single piece of feedback can be processed several times by multiple crowd members. The annotation process is structured in a way that steers the users to utilize predefined categories to be in compliance with proposed RE taxonomies for user generated feedback (Pagano & Maalej, 2013). Additionally, it allows users to add their own categories but those are only implemented in the entire system after consideration and review of the requirements engineers. Annotated pieces of user feedback can be collected after processing for further analysis by requirements engineers.

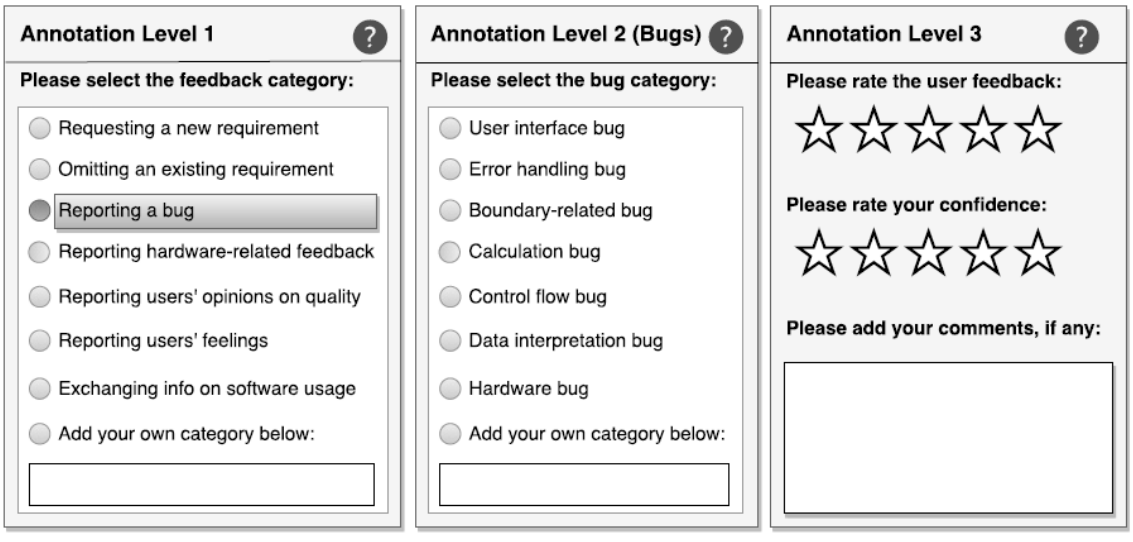


Figure 8: The CRAFT approach from the perspective of the crowd worker by Hosseini et al. (2017).

The crowd itself faces the annotation process in a different way, and can be assigned in different tiers. The first tier is more high-level and asks users to specify what type of feedback they want to annotate, basically facilitating the categorization process. The second tier is more low-level and focuses on subcategories defined for each category from tier one. This specifies what the piece of user feedback entails and what it addresses more specifically. The last tier request three final

inputs from the crowd, assigning a level of importance based on priority, specifying their confidence in the annotated piece of user feedback and allowing them to add comments.

The craft method has run a trial in the form of a small case study with 12 graduated computer science students. This trial concluded that the results of the feedback categorization were satisfactory in general, even though some participants made mistakes with the categorization process. They also found that all participants rated the importance of their feedback medium to high and had high confidence in their own annotations.

Lessons Learned

While evaluations of existing methods in practice are scarcely available, the existing methods indicate a certain level of feasibility for crowdsourcing platforms. The StakeRare method is one of the most extensively reviewed methods available and showed promising results. Nevertheless, StakeRare was tested in a limited environment (one single large-scale software project) and a context with unique characteristics (specific access control system for a single university). Therefore, the question remains how generalizable the results are. Applying a similar method in a software development organization of smaller size with less involved stakeholders may result in the loss of identified benefits of said method.

The authors of the developed methods all identify a certain set of challenges that threaten the success of their method. Assuring an acceptable level of quality and properly engaging and incentivizing a crowd of a significant size being recognized as the most dominant ones. The authors all propose different solutions that aim to either tackle or minimize the effects of these challenges but their effectiveness has to be questioned due to lack of evaluation and lack of deployment in the real world. However, requiring participating crowd members to take a test to gauge their knowledge level, as well letting multiple different crowd members complete a task separately, appear to be promising measures to stimulate a higher level of quality of the results.

2.2.6 Conclusion

The presented theory regarding crowdsourcing practices, tasks and crowds have some design implications for the to be developed crowdsourcing method. When comparing the three main types of crowdsourcing defined by Bigham et al. (2015), we can conclude that each different type comes with an unique set of benefits and challenges, which are summarized in Table 3.

Type	Leadership	Incentives	Benefits	Challenges
Directed	Directed by Requester	Extrinsic	High amount of work that will be done	Low quality results
Collaborative	Generally self-organizing	Intrinsic	No external incentives needed, high quality return	Prone to conflict Knowledge level of crowd
Passive	Separate from crowd	Intrinsic	No external incentives needed No direct work requests required	Preventing interference Requires elaborate infrastructure

Table 3: The distinguishing factors between the three main types of crowdsourcing.

In a sense, online reviews are already passively providing feedback about software gained from users by crowdsourcing means. Analyzing these online reviews and processing them into a usable set of requirements however requires a whole different approach and cannot be achieved passively. Going for the collaborative approach mitigates the challenges in incentivizing the crowd and heightens the quality of results, but can only be achieved when the crowd possesses the right amount of knowledge. Going for a more directed approach makes it relatively easier to incentivize the crowd, but may hurt the overall quality of the results in the process. In practice it is likely that a hybrid approach of both directed and collaborative method will offer the most benefits under the condition that the challenges are properly addressed.

The activities that the crowd will have to perform can be classified as both crowd labour and crowd collaboration. The tasks that have to be performed can be classified as complex tasks as it considers characteristics of problem solving, but the tasks itself are recommended to be executable as modular and as close to micro tasks as possible.

The crowd required for RE processes can be classified in the category of product/service innovation in the model defined by Erickson et al. (2012). While it does require the crowd to possess some specific domain expertise, it allows the organization to benefit from all three types of crowd value. Using crowdsourcing techniques for the innovation of a product provides access to the benefits offered by a large, diverse group of workers and their high amount of distributed knowledge.

Regardless of the exact form, crowdsourcing does offer added value for the RE discipline. It enables the possibility for feedback-based requirements engineering as it can process a continuous flow of user feedback for utilization in development process, something that practitioners have struggled with in the past (Maalej et al., 2016). When implemented correctly a plethora of benefits can be reaped by software development organizations. These potential benefits are summarized below.

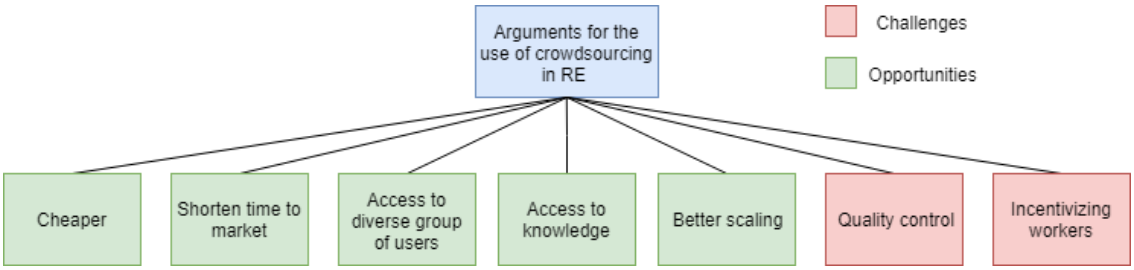


Figure 9: Summary of the arguments for the utilization of crowdsourcing in RE practices from the crowdsourcing perspective.

1. **Cheaper:** Utilizing forms of crowdsourcing is cheaper than complete outsourcing or having own dedicated employees (Adepetu et al., 2012).
2. **Shorten time to market:** The crowd can support the requirements engineers and reduce their workload, which in turn can shorten the time to market of the product (LaToza et al., 2013).
3. **Access to diverse group of users:** Properly acquiring and training a crowd may provide access to a wide diversity of actual and potential users, that can be utilized in the long term during the life-cycle of the application (Hosseini et al., 2014).
4. **Access to knowledge:** Considering online reviews as workable input to be processed by crowdsourcing practices offer previously untapped knowledge or expertise (Groen, Kopczyńska, et al., 2017).
5. **Better Scaling:** Crowdsourcing methods scale better than traditional elicitation methods, which allows for the acquisition and processing of huge amounts of data (Finkelstein & Lim, 2012). This has potential to better align RE practices with agile development methods (Mao et al., 2015).

These benefits can only be reaped if the accompanying challenges in crowdsourcing are dealt with accordingly. Quality assurance and incentivizing a crowd are perhaps the biggest obstacles. The ideal crowd therefore appears to require a certain level of knowledge and expertise, as well as has some kind of emotional investment or other intrinsic motivation in the process. Techniques that aim to achieve these two factors are available, but should be approached cautiously due to the lack of validation and application in practice.

2.3 Natural Language Processing

The rise of social media platforms has significantly increased the volume of feedback from software users (Dalpiaz & Parente, 2019). While this increase in volume offers new opportunities, it requires techniques that are able to cope with this amount of data. So far only crowdsourcing has been discussed as one solution for this predicament, but other techniques exist as well.

A different way to process a large amount of user generated is by applying data natural natural language processing techniques. Natural Language Processing (NLP) is defined as *...a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications* (Liddy, 2001). This section describes the core principles of NLP techniques, illustrates the role of NLP in RE practices and highlights the current challenges within the field.

2.3.1 NLP Basics

The aim of a linguistic science is defined as the ability to characterize and explain all linguistic observations circling around us, in conversations, writing, and other media (Manning, Manning, & Schütze, 1999). NLP is one method for the analysis of written language and as stated in the definition by Liddy (2001), aims to reach a level of language processing that is as human-like as possible (Manning et al., 1999). NLP is therefore often classified as a form of Artificial Intelligence, as it aims to reach the same capabilities of language interpretation that a human possesses. As claimed by Liddy (2001), NLP techniques are mostly only applied with the purpose to achieve a set of goals or to complete a set of specific tasks.

NLP techniques mostly strive to complete a specific goal or task, so the application of the techniques itself is not usually considered the final goal. NLP is usually applied to process a set of (mostly textual) linguistical content, so that it can aid in reaching said final goal. Liddy (2001) defined four categories of NLP processing activities that illustrate how linguistical content can be processed.

1. **Paraphrasing an input text:** Processing the initial input so that it can expressed in a different way, to improve the overall clarity of the text.
2. **Translating a text into another language:** Transforming the input text into a different language, with as little loss from the original meaning as possible.
3. **Answering questions about the contents of a text:** Gaining sufficient understanding of a text by automatic means, so that questions about the input text can be answered.
4. **Drawing inferences from a text:** Deducting conclusions from a text about what is not explicitly mentioned by the writer, but does match the original intentions of said writer.

The capabilities of the NLP field have advanced substantially over the years, which are mainly attributed to four specific reasons by Hirschberg and Manning (2015). Firstly, the vast increase in availability of inexpensive computing power enabled the processing of more complex tasks. Secondly, the Internet provided access to an enormous set of linguistical data in a digital form, readily available for research purposes. Thirdly, effective machine learning techniques were developed that aided in the analysis of this newly available data. Lastly, advances in the field of linguistical studies resulted in a much better understanding of the human language and its deployment in social contexts. The combination of these factors created an environment where other fields could benefit from and thrive, with one of them being requirements engineering.

2.3.2 NLP for RE

Natural Language processing has been applied to requirements specification from very early on (Ryan, 1993) as natural language is the predominant notation for documenting and specifying software and system requirements (Kassab, Neill, & Laplante, 2014). Applying NLP techniques in a field where natural language has such a dominant presence seems nothing but obvious. Requirements documentation for complex systems can grow to a extend that it that becomes hard to manage due to its size. NLP techniques can assist a reader of these documents to identify the concepts used by the writer and the relations between them (Kof, 2004).

While natural language is the dominant notation for requirements, it has shown shortcomings to function as a precise, concise and unambiguous medium (Berry, Gacitua, Sawyer, & Tjong, 2012). Throughout the years, NLP tools have been developed that aid in detecting defects and shortcomings of natural language for specification purposes. The simplest category of these tools merely try to detect formatting and syntactic format violations, and therefore identify bad practices in the specification process of requirements. However, more advanced techniques also exist that generate models such as class diagrams or tools that create traceable links between descriptions of requirements and artefacts used in the development process (Berry et al., 2012).

Especially uncovering the relationships between different concepts within requirements documentation has been a point of interest for the field. Creating traceability links aims to aid software engineers in improving the understanding of the relations and dependencies among software artefacts (Y. Zhang, Witte, Rilling, & Haarslev, 2006). Attempts have been made in the past to generate these trace links between customer wishes and specified requirements through automatic means, with the aid to speed up the development process (och Dag, Gervasi, Brinkkemper, & Regnell, 2004). In practice, it turned out that this method reduces the required effort for large scale requirements management by 66%, indicating the merit of similar NLP techniques in practice (och Dag, Gervasi, Brinkkemper, et al., 2005).

Additionally, the rise of social media offered new possibilities to acquire linguistical data sets for research purposes as previously mentioned. The surge in popularity in opinion-rich resources such as online review platforms opened up new opportunities to discover what people think about a certain subject (Pang, Lee, et al., 2008). The emergence of opinion mining and sentiment analysis techniques fostered RE researchers to develop approaches that mine requirements for the purpose of software product improvement (Qi, Zhang, Jeon, & Zhou, 2016). The latest advances within the opinion mining field shifted the focus of studies into short-length texts, spam detection and contradiction analysis in order to cater it to reviews found in mobile app store user reviews (Genc-Nayebi & Abran, 2017).

2.3.3 Challenges and Shortcomings

The use of NLP techniques to benefit RE and software engineering practices have shown their usefulness and added value, but their application in practice appears to be hindered due to encountered challenges originating from different sources. While gaining access to workable datasets by applying automatic scraping and mining techniques on opinion-rich platforms are widely available, the possibilities of the techniques for fully automatic analysis are limited to mostly only being able to classify opinions as positive or negative (Penalver-Martinez et al., 2014). Gaining further insights often requires the technique to become semi-automatic by including the intervention of an expert to obtain their results. Furthermore, most existing techniques only address these limits by only focusing on simple tasks or making trade-offs between the precision and recall of the tool (Dalpiaz, van der Schalk, Brinkkemper, Aydemir, & Lucassen, 2018), further limiting their usefulness in practice.

The nature of the acquired data from online sources also introduces challenges that limit the effectiveness of analysis by NLP techniques. As previously stated, natural language is inherently ambiguous (Berry et al., 2000), which is something that automated NLP techniques struggle to cope with effectively (Dalpiaz, Van Der Schalk, & Lucassen, 2018). Additionally, online sources such as app stores provide user generated feedback on such a scale, that the sheer numbers push the analysis towards automated means due to the amount of effort required to analyze them manually (Groen et al., 2018). This is in contrast with the persuasion that RE is a fundamentally human-centred process (Bourque et al., 2014) and breaks away from the call to develop RE tools that keep the human factor intact (Berry, 2001).

Furthermore, the noise that is inherently present in user generated feedback sources (Williams & Mahmoud, 2017) further impact the performance of elicitation techniques. As N. Chen, Lin, Hoi, Xiao, and Zhang (2014) illustrated, the manually established ground truth review in their review found that only 35,1% of reviews possessed relevant informative information that can aid developers in their development processes. Resolving noise is further complicated by the lack of

meta-data about the users that provided the feedback on online sources. This prevents analysts from including demographic data into their analysis to derive contextual information. This is further complicated by the fact that the user-developer communication is unidirectional, which prevents the development team to reach back the users and ask for clarifications and context information (Dalpiaz & Parente, 2019). Automated systems have been introduced to identify fake reviews and reviews littered with spam to evaluate their usefulness, but these systems are limited and not yet mature (Genc-Nayebi & Abran, 2017).

Moreover, sets of gathered online reviews are notorious for containing conflicting opinions, which are hard to deal with for opinion mining techniques (Pang et al., 2008). Effects from this phenomena are highly impactful on the performance of fully automatic opinion mining techniques as they are mostly only capable of categorizing reviews as positive or negative. As reviews can contain multiple positive and negative statements, it is challenging to trace these statements to the exact feature that they refer to (Penalver-Martinez et al., 2014). Additionally, the reviews themselves are often provided in other languages than English. Combined with the notion that text mining applications become less usable when other languages are used (Hosseini et al., 2017), further restrict their applicability in practice as it disqualifies a set of reviews from inclusion. This, in combination with the notion that online ratings are typically generated by a fraction of users (Gao, Greenwood, Agarwal, & McCullough, 2015), may hurt the validity of the results as it is possible that a relevant user group is not represented in the analysis.

Lastly, research on eliciting requirements from a large number of online reviews using automated means has indicated a tendency to focus on functional aspects (Groen, Kopczyńska, et al., 2017). This leaves quality aspects of a system underexposed, while statements addressing them appear to be widely available in online reviews (Lu & Liang, 2017).

2.3.4 Conclusion

This section discussed the basics of NLP techniques, their practices in the RE discipline and the current challenges in the rapidly evolving landscape. While NLP and opinion mining practices and techniques have shown their merit for RE practices, a myriad of challenges limit their effectiveness for user feedback gathered from sources such as online review platforms. These challenges are summarized below in Figure 10.

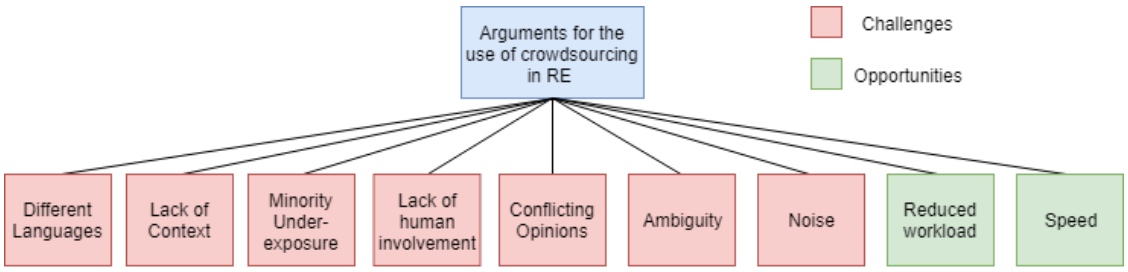


Figure 10: Summary of the arguments for the utilization of crowdsourcing in RE practices from the NLP perspective.

1. **Ambiguity:** Natural language is inherently ambiguous which is hard to solve without clarification of the respective author.
2. **Lack of human involvement:** The sheer volume of user generated feedback steers practitioners towards the use of fully automated means of analysis, which is in contrast with the persuasion that RE is a human-centred activity fundamentally.
3. **Noise:** The high level of noise and spam in online reviews are hard to filter without human intervention.
4. **Lack of context:** The lack of available meta-data about users and the unidirectional way of communication prevents the analysts of online reviews to place their results into context.
5. **Conflicting opinions:** The possibility of conflicting opinions within or between online reviews are hard to process by one-dimensional opinion mining techniques.

6. **Different languages:** Online reviews are provided in languages other than English but text mining applications become less usable when other languages are applied.
7. **Minority underexposure:** NLP techniques are most advanced for the English language, leaving out potentially important feedback from other language sources. Combined with the idea that online reviews are already mostly provided by a small group of users, the threat arises that unconventional opinions get buried by automated processes.

While NLP techniques certainly have an established beneficial role within RE practices, they currently appear to be ineffective to deal with the challenges introduced by user feedback platforms such as mobile app stores. The desire to include online reviews into development processes introduce new challenges that are hard to tackle by techniques that rely on fully automated means only. These aspects build towards the case for more human-centered RE solutions, or a mix of both, such as crowdsourcing to keep the human factor into a fundamentally human focused practice intact. However, the desired crowdsourced RE solutions requires the challenges introduced with the involvement of the crowd to be dealt with adequately.

2.4 Industry Practices: Insights from an Interview

To gain insight into RE practices in the real world, an interview was conducted at a consultancy company in The Netherlands. The company in question is Deloitte ltd. and offers consultancy services for accountancy practices and other types of financial advisory, risk assessments and company strategies and technological advancements. Among these areas, a branch within Deloitte offers services that aid software companies companies to determine their strategies and to further develop their products.

With the main focus on software development, the goal of the interview is to identify their current RE practices, gaining insight into their practices that deal with user feedback and to identify reoccurring challenges that they observe within their field. Lastly, the aim was to explore their vision on crowdsourcing practices and to see whether these would have a place in the real world, in their opinion.

An interview protocol was constructed to guide the interview itself and to provide structure for the processing of the results.

2.4.1 Interview Protocol

General Questions

1. What is your background?
2. What is your current role within the organization?
3. Could you explain something about the roles and the goals within your department?
4. How many and what kind of clients do you handle?

Requirements Engineering Practices

1. What are the general practices used for the elicitation, analysis, prioritization and specification of requirements?)
2. From what sources do you gather user feedback/requirements?
3. Have these practices changed over the years?
4. Do you expect any notable changes within the near future?
5. Do you know of any challenges in the industry regarding RE. If yes, could you describe them and explain how they are dealt with?
6. Do you recall any notable successes/failures happening?

More detailed explanation of current thesis project and the focus on crowdsourcing in RE.

Crowdsourcing

1. Do online reviews of software currently play a role within the RE process? Why?/Why not?
2. Do you personally have any experience with crowdsourcing at all?
3. Do you recognize the potential of the utilization of crowdsourcing?
4. Do you see possibilities for a method resulting from this thesis project?

2.4.2 Interview Results

The interview took place on the 23rd of January 2019 at the Deloitte office in Amsterdam. The voluntary interviewee for this interview was Ernst Fluttert, an senior consultant from the digital technology department of Deloitte. His expertises are in the areas of digital architecture and agile practices. The following parts summarize and paraphrase the most important answers for each question from the interview protocol.

General Questions

1. **What is your background?**

   Ernst graduated at Twente University for his bachelor's degree in Business and IT and his master's degree in Computer Science. He acquired experience as a .NET developer before ending up at his current position in Deloitte. Currently he is a team leader for solution architecture with his expertise being around team growth.

2. **What is your current role within the organization?**

   Ernst is a professional agile coach certified in Scrum, Kanban and SAFe. His current activities involve coaching and leading teams to work iteratively in teams for software development. He is also considered a meta-expert on solution architecture, involving him in many project that require his expertise to review new proposals. This mostly considers large project for large companies often aimed at implementing updates for legacy systems. In addition to this, he provides Agile and Scrum training for clients that want to implement a more agile way of working into their practices.

3. **Could you explain something about the roles and the goals within your department?**

   The department where Ernst is working is called Deloitte Digital, which is part of the entire Technology branch. His daily activities are classified under the role of digital engineering. The creatives and the strategists for advisory are the other two roles present within the department. Even with these three classes, similar people with similar activities and background as Ernst work in this department, all having their main focus based on working for clients and fulfilling their desires. The exact roles that employee play on a daily basis are dependent on the current clients.

4. **How many and what kind of clients do you handle?**

   Clients of Deloitte are mostly the largest companies present in the Netherlands, as SMEs generally do not quickly approach Deloitte. The number depends on the size of certain projects that are currently being handled.

Requirements Engineering Practices

1. **What are the general practices used for the elicitation, analysis, prioritization and specification of requirements?)**

   The requirements elicitation often starts from plain business values or mission statements. As they come in as a third party, they have to become acquainted with what the business wants to achieve, so they can adhere to their business objectives. After this, they want to know where the business is currently at, by looking at their current landscape. When available, a roadmap can provide a lot of information to establish the right way forward. They therefore also often consider business and technological requirements in addition to functional and non-function requirements, so their solution will fit the existing infrastructure and still aligns with the strategy of the business. This is often considered the basis for their RE practices.

   To acquire more detailed requirements, workshops with the purpose to elicit more detailed

information. This is mostly an organic process, but depends on the nature of the organization. Government institutions often require a lot more structure and time as they are much more dependent on the internal hierarchy within such an organization. Regardless, the elicitation takes place just by asking questions and by having conversations and discussions with people from the organization. These conversations are mostly held with people from upper management, but other employees also get involved when details have to be filled in. No internal guidance structure for these processes are present at Deloitte, decisions are often made on the go based on the perceived information required to complete the project.

In these workshops, discussion often starts off from feature level because that is most tangible for the client. The technical details are often filled in later based on these requirements. The features are split into so called epic requirements on a higher level that are often expressed in stories. The details for the exact interactions are often filled in at a later stage with users or with someone that represents the desires of the users or employees that will be using the system. A distinction will be made between functional and non-functional, but is not seem as very important. The main focus is always to get the system working, as there is always time after to fill in the details and to work in other requirements. It can be a challenge to properly with the different kinds of requirements and to prioritize them correctly.

2. **From what sources do you gather user feedback/requirements?**

   The main source of the system requirements often come from upper management layers that participate in these workshops, but other roles come into play when necessary.

3. **Have these practices changed over the years?**

   Yes, the practice of dealing with requirements have changed over the years. In the past, a more waterfall approach has been utilized where all possible requirement are specified before the start of the development. This often resulted into the discovery that not all requirements were correct at a very late stage of the project, causing delays and cost increases. Compared to the past, a shift towards a more agile way of working has been introduced that is focused on small and short iterations. This is the trend that I have observed, being it by using Scrum, SAFe, Kanban or any other format. DevOps is maybe the most recent hype, but the overall trends is working in short cycles.

4. **Do you expect any notable changes within the near future?**
   I somewhat expect cycles to become even shorter, due to an increased desire to reduce the time to market. Basically everything should be going quicker and development time should be reduced, as it becomes harder and harder to expect events in the future. This requires carefully monitoring parts where development can go wrong and to prevent errors or mistakes to be caught as early as possible to diminish their negative effects. This can be hard sometimes, as change in general can be tough to deal with.

5. **Do you know of any challenges in the industry regarding RE. If yes, could you describe them and explain how they are dealt with?**

   Working in short cycles introduces a couple of challenges. Generally, establishing the requirements and the deployment for production take up the most time. When cycles become shorter, these activities have to be performed more often, so any automation or alternatives that speed up these processes are welcome. Regarding the requirements specifically, higher priorities have to be laid on crucial parts and the core functionalities of the system. The requirements are used to establish acceptance criteria as well, so if those are not present then there is less time available for testing.

   All these activities that have to be performed in incrementally less time unfortunately sometimes hurt the quality due to the rush for a single deadline, which hurts the planning for the following sprints. I think that this is the biggest challenge for the agile way of working, as incomplete stories or requirements negatively affect other sprints due to lagging behind

previous events.

6. **Do you recall any notable successes/failures happening?**

   I have observed multiple times that a team is 100% focused on just the deadline, resulting in the preparation for the next phases being neglected. This always severely impacts their effectiveness and productivity in these phases and hinders the communication about crucial aspects. This has occurred multiple times between multiple different clients. I do however also have a good example at a government institution that wanted a specific feature that handled documents. They correctly planned multiple workshops for multiple sprints, that quickly discovered that they underestimated the scope of the project. As this error was caught early, the effects were limited and the project was finished relatively quickly and within budget.

Crowdsourcing

1. **Do online reviews of software currently play a role within the RE process? Why?/Why not?**

   This highly depends on the project itself, as larger companies often have their own issue tracking system to acquire feedback. On the other side, smaller companies that we deal with are mostly not at all near a stage where direct user feedback becomes very relevant. In a sense, feedback from users often are neglected a bit because they are not the business and technology requirements that we need, with the exception of bugs.

   It does however occur that improving user interactions is a business objective, so that is where user reviews are included in the process. This has happened in the past, but the reviews had to be analyzed manually. The resulting quality of this was alright, but not perfect. They are nevertheless hard to deal with because users mostly refer to aspects that they directly see or encounter. We therefore consider feedback from sources where there is a mutual interaction as a bit more valuable. Blijft wel lastig omdat gebruikers vooral reageren op wat ze wel zien.

2. **Do you personally have any experience with crowdsourcing at all?**

   I am familiar with the concept, but I do not have personal experience with it at the moment to be honest.

3. **Do you recognize the potential of the utilization of crowdsourcing?**

   In the way to you have explained your currently proposed method, I do think it has potential. In my experience, humans can be quite capable in recognizing the reviews that have garbage quality. Even if it results in a set of reviews that removed the reviews of the worst quality, I think this would help experts to find the more valuable feedback more easily and with less effort. It does however gives only feedback on the current version of the app and what the users currently interact with and does not take into account solutions that are already in the pipeline. I think you should also consider NLP techniques that can already remove the worst reviews before subjecting it to humans. I think that would make it a bit more efficient as humans are generally quite expensive.

4. **Do you see possibilities for a method resulting from this thesis project?**

   I think i can see the potential mostly for use case based analysis, where it is required to place the feedback in its precise context, for instance related to a specific current version of an application. Additionally, to be fully useful, I would say that it should be capable of separating the feedback in different themes of feedback instead of only filtering out the bad quality ones. I can think of some possibilities, but I must be honest and say that I do have some concerns about the feasibility.

2.4.3 Conclusion

The interview with Ernst Fluttert at Deloitte further substantiates the observed general shift from waterfall practices towards more agile ways of working. These agile methods require progressively more and smaller work cycles that have to be completed in less and less time. For each of these cycles, the requirements engineering practices are one of the activities that require relatively a lot of time due to planning activities for other sprints. Due to the existence of this bottleneck, this process is desired to be sped up as they are needed for correct project planning. This is marked as one of the biggest challenges for methods that provide a more agile way of working and thus providing argumentation for the need of quicker elicitation methods. This provides perspective for the introduction of alternative methods, that for instance utilize crowdsourcing to speed up some parts of the process. However, doubts have been raised about the feasibility of such methods so a thorough investigation into crowdsourcing practices is warranted.

2.5 Crowdsourcing Dynamics

Earlier we discussed the benefits of the utilization of crowdsourcing methods and what challenges threaten the possibility to reap these benefits. Developing a crowdsourcing method that conserves the human-factor in RE practices appears attractive, but requires the mitigating of potential detrimental effects originating from common challenges and pitfalls of crowdsourcing techniques. The upcoming sections explore these dynamics and presents existing mechanisms found in literature and existing crowdsourcing techniques that offer possible solutions for these common challenges and pitfalls.

2.5.1 Quality Control

As discussed earlier, the challenge of quality assurance of acquired results appears to be a trend traversing all forms of crowdsourcing techniques (Vukovic & Bartolini, 2010). To improve the usefulness of the acquired results, it is desired that mechanisms are in place that control the quality generated output by a crowd. Allahbakhsh et al. (2013) propose a taxonomy for quality in crowdsourcing systems derived from observations in practice. Their taxonomy represents quality control attributes that have to be considered to make sure that the outcome fulfills the requirements of the work requester as best as possible. As shown in Figure 11, a division is made between quality attributes addressing the crowd worker and of the tasks to be performed.

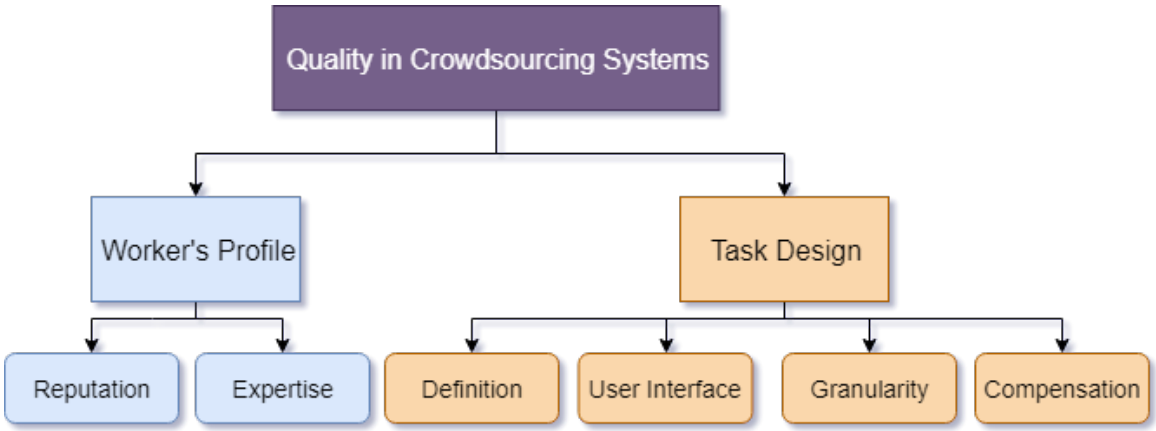


Figure 11: Taxonomy for quality in crowdsourcing systems (Allahbakhsh et al., 2013).

The workers profile addresses the crowd workers themselves, as their abilities and commitment affect the created outcome. Having measures about the exact abilities and the quality of a worker allows for the creation of an indication about the created results of the work. The two attributes proposed by Allahbakhsh et al. (2013) take into account the reputation and the expertise of each worker involved in a specified set of tasks.

1. **Reputation:** The reputation of a crowd worker is built as a community wide metric, reflecting the trust relationship between the worker and the work requester. It should function as an indicator for the probability that the requester will receive a high quality contributions of said worker, and as an indicator for the reputation that a worker has compared to other workers. Reputation scores can be built around explicit and implicit feedback. Completed work can be explicitly reviewed and scored based on direct reviews, ratings or ranking by the work requester or other members from the community. The alternative is to implicitly derive the quality of a contribution based on required corrections of work in the past.
2. **Expertise:** The expertise of a crowd worker demonstrates their capability, and how suitable they are for a particular task. The first indicator of expertise are the credentials of a worker. Credentials refer to any proof of concept that a worker has previously reached a certain level of knowledge or show that the worker possesses a certain set of skills. The second indicator for expertise is experience and refers to the knowledge and skills that a worker has acquired while working with a system or a technique. The number of completed tasks and the time frame of their contributions give an indication on how experienced a particular worker is and

therefore give an indication about their expertise. The overall expertise of a worker can be increased by coaching, guiding and supporting the workers in doing their work.

Not only the performance of crowd workers affect the quality of the generated results, but the way that the tasks are designed also have an effect. The tasks that are presented to the workers for execution can be divided into different components that all have an affect on the ability of a worker to complete them.

1. **Definition:** The task definition refers to the way that a task is presented to the crowd. A task in its entirety should hold a description about its nature and goals, completion criteria and any limitations such as time constraints. Additionally, qualification requirements can also be set that state when crowd workers are eligible to participate. The eligibility criteria can be set based on worker reputation and experience, but also on other geographic or demographic characteristics of the worker.
2. **User Interface:** Refers to the means that a crowd worker has access to the to be performed tasks. A user friendly interface that is perceived to be easy to work with has the chance to increase the resulting quality, as it allows user to focus on their work more easily. However, interfaces should not be too simple and easy so to reduce the chance of exploitation by malicious workers. A fine balance should be sought after to achieve a UI that is intuitive to work with, advanced enough to facilitate the performance of the tasks, but not too complex to cause delays.
3. **Granularity:** Whether the work to be done are simple or complex tasks affect the quality of the outcome. Simple tasks are generally short and self-contained and require little expertise to be solved. Complex tasks usually require more time, effort and expertise, thus making them less appealing for the workers. For the sake of quality, complex tasks should be broken down into smaller, easier to solve sub-tasks. The set of sub-tasks should be defined as a structured workflow to provide structure and to offer guidance with the processing of the results.
4. **Compensation:** The policy for compensation influences the motivation of the workers and therefore the quality of their performance. We already discussed the discrepancy between intrinsic and extrinsic motivations, but it is important to note the balance between the two. Intrinsic motivation generally has a more significant positive effect on the results, but only for people that are honestly concerned with the content of the work. Nevertheless, extrinsic motivation in the form of monetary rewards are the most common kind of rewards. General lessons learned from monetary rewards are that the sheer amount does affect the amount of work done, but does not necessarily offer a direct benefit for the quality of the results. The way the rewards are distributed also affect the quality, as there are choices to offer the rewards for the completion of individual subtasks, but also additional rewards for the completion of an entire complex task. As incentives are one of the most important pain-points in crowdsourcing practices, they will be discussed more extensively in the following sections.

The aspects discussed above give an indication about all the factors that can be managed to influence the quality of results produced by crowdsourcing techniques. However, how these mechanisms work when they are applied in practice is still unclear and rather vague for now. In addition to the taxonomy presented above, Allahbakhsh et al. (2013) gathered and summarized mechanisms that address these quality aspects in existing methods that rely on crowdsourcing techniques. To make these possible mechanisms more tangible and to offer a starting point for the actual construction process of the proposed technique, these quality-control approaches are summarized below.

Allahbakhsh et al. (2013) make a distinction between quality-control approaches for design-time and run-time. They differ from each other based on the point of at what time they should be considered and where they are the most effective. As their labels indicate, design-time approaches have to be considered during the design and construction phase of a crowdsourcing technique with the aim to preemptively improve the quality of the results. This can be achieved by the following approaches:

1. **Effective task preparation:** Stemming from the idea of defensive design to make the tasks that the workers have to perform as fault proof as possible. This can be achieved by

providing unambiguous descriptions of the tasks at hand, preventing that cheating is easier than properly finishing the task and to define evaluation criteria beforehand.

2. **Worker selection:** Including a selection process to allow a specific section of workers to contribute to the cause. Selections can be made on based on reputation, credentials, demographic characteristics or a mixture depending on the required minimum level of quality.

On the other hand, run-time approaches are implemented to increase the quality of the results immediately after they are contributed by crowd members. The following approaches have the potential to achieve this:

1. **Expert review:** Having domain experts at hand to actively review and check the quality of contributions made from crowd workers.
2. **Output agreement:** Accepting contributions to be correct when all contributions provided by the crowd workers independently reached similar conclusions.
3. **Input agreement:** A check based on mutual understanding of the description, purpose and goal of a task at hand between individual crowd workers.
4. **Ground truth:** Comparing contributions with an established golden standard
5. **Majority consensus:** Accepting that the judgment of the majority of crowd workers provides the highest level of quality.
6. **Contributor evaluation:** Evaluating a contribution based on the previous performances, reputation or other types of credibility of an individual crowd worker.
7. **Real-time support:** Having supporting staff available to support the crowd workers in completing their tasks with increased quality.
8. **Workflow management:** Defining a specific workflow so it provides structure to monitor the progress of complex tasks.

These quality-control approaches are not mutually exclusive. It is possible to implement multiple approaches or variations on them so that they improve the overall quality of the results in an additive way and strengthen each others effects. Nevertheless, the feasibility of each of the possible quality-control approaches has to be considered during design processes.

2.5.2 Crowd Management

As previously discussed, to improve the effectiveness of crowdsourcing techniques, it is important to match the characteristics of a crowd to the purpose and goals of the technique itself (Bogers et al., 2010). A crowd can bring a wide variety of usable skills and knowledge to the table (Mumford, 2003), so it is important to find the right dynamic between the crowd and the to be constructed technique to make optimal use of their capabilities. In order to facilitate this, the decision making process for the way that the participating crowd is involved and managed in the technique has to be approached carefully.

The first step is gaining more insights in the way that the crowd is involved in crowdsourcing practices. In their taxonomy, Geiger, Seedorf, Schulze, Nickerson, and Schader (2011) identified four dimensions at where crowdsourcing practices can differ on the way that they involve the crowd. These dimensions are shown in Figure 12.

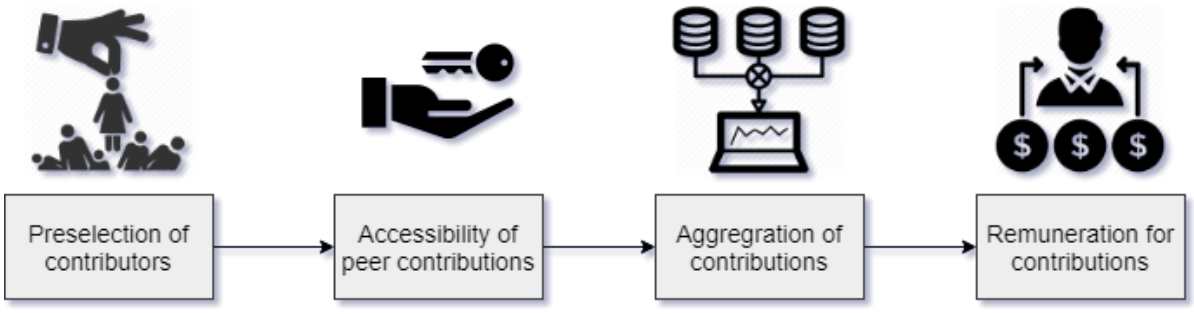


Figure 12: Taxonomy for crowd involvement in crowdsourcing processes (Geiger et al., 2011).

1. **Preselection of contributors:** The definition of crowdsourcing given by Howe (2006) states that crowdsourcing practices concern an open call to a large and undefined group of people. However, this does not mean that practitioners are not allowed to shape the characteristics of the crowd that they are recruiting. Geiger et al. (2011) state that most crowdsourcing practices strive to benefit from as much diversity and scalability as possible and therefore do not aim to restrict the participation of contributors. However, it is not uncommon to exclude contributors when they do not adhere to rules, codes of conduct or required quality standards. Depending on the purpose of the technique, Geiger et al. (2011) identified two different selection and filtering techniques applied in practice.

   (a) **Qualification based:** Allowing contributors to participate when they have demonstrated that they possess a certain level of knowledge or skills.
   (b) **Context-specific:** Allowing contributors to participate only when they originate from a specific context or possess certain characteristics. For example: employees from a specific organization, a group of users of specific software or any other demographic characteristics like age or location.

   It it possible for crowdsourcing techniques to apply principles of both qualification based and context-specific selection approaches. However, this introduces the risk of ending up with a selection policy that is too restrictive and filters out too many potential crowd contributors, resulting in the loss of benefits gained from a large and diverse group of people.
2. **Accessibility of peer contributions:** Peer accessibility refers to the dynamics between the crowd contributors themselves. To what extent they have access to each others contributions can vary a lot depending on the purpose of a crowdsourcing technique. It is possible to be as restrictive as possible due to privacy concerns or preventing that contributors influence or copy from each other. On the other hand, for the stimulation of collaboration it might be more beneficial to not separate the contributors too much. Geiger et al. (2011) defined four characteristics within this accessibility dimension.

   (a) **None:** The most restrictive level that isolates contributors and their contributions from each other. Contributors cannot see, reuse, complement or even react to individual contributions.
   (b) **View:** Individual contributions are made visible to any potential contributors for viewing purposes only. Contributions can be viewed for for instance inspiration purposes, but there are no means for commenting or reacting available.
   (c) **Assess:** Contributors have viewing access to contributions of other crowd members and explicit mechanisms are in place that stimulate reactions on said contributions. It allows peer contributors to react and express their opinions on individual contributions. These mechanics are often placed to help filter a large amount of contributions or to get an indication of the overall quality of contributions.
   (d) **Modify:** The highest and most open level of peer accessibility. Here it is possible for contributors to edit or delete contributions made by peers, with the main purpose of correcting, updating or improving them. This has the potential to improve the overall quality of all contributions due to the factor of collaboration involved, but can be quite hurtful as well by crowd members with malicious intent.

   The four characteristics within the peer accessibility dimension show in which different ways the dynamics between crowd contributors can be shaped. It is important to carefully consider the purpose of a crowdsourcing technique and to find a matching crowd dynamic that has enough transparency to improve the quality of contributions but to keep the chance of harmful practices from occurring to a minimum.
3. **Aggregration of contributions:** The number of contributions provided by the crowd directly scales with the amount of crowd workers involved in a crowdsourcing technique. Especially for techniques that utilize a large number of crowd workers, the number of provided contributions can quickly grow to a size that makes it harder or more work intensive to process them all. The strategy to process all the contributions by the crowd therefore has implications for the effectiveness of the crowdsourcing technique. Geiger et al. (2011) defined two possible approaches for the processing of provided contributions.

(a) **Integrative:** An approach where all contributions are pooled together and combined to determine the final outcome. Only contributions that fail to adhere to the most basic rules and quality standards are filtered out.
(b) **Selective:** A more competitive approach where individual contributions are compared to each other and where the best contribution, or a set of contributions, with the highest quality are selected. Selection criteria can be based on the direct desires from the work requester or based on the collective opinion from the crowd, derived from their assessments and ratings.

Which of the two approaches offers the best fit for a particular crowdsourcing technique depends on the size of the crowd, the amount of contributions made and the resources available to process the contributions in order to derive the final results.

4. **Remuneration for contributions:** We discussed compensation for contributions made by crowd workers before and shown that they function as an incentive to motivate the crowd workers to perform to the best of their ability. While monetary rewards appear to be the most prevalent ones in crowdsourcing practices, other types of rewards are just as valid. Nevertheless at their core principles, rewards can be divided into three categories as defined by Geiger et al. (2011).

   (a) **None:** No direct rewards or compensation are offered. The crowdsourcing technique is dependent on voluntary contributions provided by crowd members.
   (b) **Fixed:** All contributions that adhere to the set of rules and quality standards receive the same amount of compensation regardless of their value to the final outcome.
   (c) **Success based:** Contributions will be rewarded depending on the individual value that it provides towards the final goal of the crowdsourcing technique. Contributions that are reviewed and identified as more valuable will receive a larger compensation than less valuable contributions. Value of contributions can be derived from the degree of complexity of the completed task, the level of quality of said contribution or the amount of insight in offers towards the final solution or goal of the technique.

The taxonomy of crowd involvement for crowdsourcing practices defined by Geiger et al. (2011) show in what ways the crowd can be involved and rewarded for providing contributions and indicate what the possible dynamics between crowd members could be. Crowdsourcing techniques rely on contributions made by their participating crowd members, so it is important that characteristics of the technique itself match with the desires, incentives and provided knowledge and expertise of the crowd.

As a closing word, regardless of the exact dynamics and the way that the crowd is involved, it requires the management and organization of a group of people. For complex tasks, one possible solution to coordinate a group of knowledgeable people is by treating them as flash teams (Retelny et al., 2014). Flash teams aim to structure the problem solving process via structured collaborations between experts from the crowd. This is achieved by clearly defining the work to be done into clearly distinguishable blocks of work, that each have clearly defined inputs and output. The flash teams are the workers of the crowd that are recruited to match with the characteristics with the tasks within a block of work. Workers can be recruited based on their knowledge level and when they match, be allocated to a specific flash team to work on a specific block of work. Ideally, the workflows within the blocks are structured in a way that the output is machine-readable so that it can be used as input for other processes, but this can be viewed as an optional benefit and is not a necessity.

2.5.3 Teaching a crowd

The people who contribute to crowdsourcing techniques might have different levels of skills and expertise that are sometimes insufficient for doing certain tasks (Quinn & Bederson, 2011) Additionally, ill-defined crowdsourcing tasks that do not provide workers with enough information about the tasks and their requirements can also lead to low-quality contributions from the crowd (J. J. Chen, Menezes, Bradley, & North, 2011). It is therefore important that a crowdsourcing technique has means available to teach the contributing crowd about the purpose, the goals and the general context of the processes of the work to be done.

One approach to teach crowd workers about requirements comes from the concept of active learning (Silberman, 1996). This approach has been widely recognized as a valuable approach for the teaching of students for different concepts by actively involving them (Johnson & Johnson, 2008). Active learning approaches have been developed for multiple different domains, with computer science being one of them (Hazzan, Lapidot, & Ragonis, 2015). The guide that Hazzan et al. (2015) propose for the application of active learning in the computer science domain consists of four different stages. These stages are shown in Figure 13.

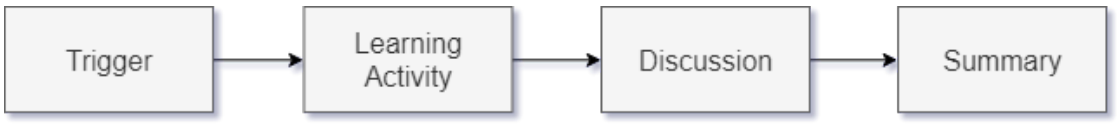


Figure 13: The four phases of actively learning approaches within the computer science domain by (Hazzan et al., 2015).

1. **Trigger:** Has the objective to introduce a topic and to sketch it in a way that is appealing to the audience. A trigger should establish the foundation for learning and should have the potential to raise a wide array of questions, dilemmas, attitudes, and perceptions. It can be in the form of a description of a relatively complex situation to introduce a new topic and to put students into a situation where they have to deal with unfamiliar circumstances.
2. **Learning Activity:** As a follow up on the previously provided trigger, presenting activities that actively guide the student through the situation of the presented trigger.
3. **Discussion:** Facilitates the refinement and corrects the students understanding of the newly introduced concepts where possible. Here, the instructor should highlight the most important ideas and work towards the construction of the desired professional perception and understanding of the introduced concepts.
4. **Summary:** Summarizing the introduced concepts and their explanations, providing clear descriptions of the purpose and the goals and to give insight in the desired answers or problem solving characteristics that are required of the student.

The role of the instructor changes throughout the progression of the four phases. During the trigger phase, the trigger should be constructed and presented carefully and thoroughly as it functions as the basis for the entire model. During the learning activity, the instructor should provide guidance through the activities and to encourage students to deepen their thinking about the concepts presented to them. During the discussion phase, the instructor should foster reflection processes and should highlight the most important facets of the presented concepts. In the final phase, the instructor should provide a logically organized summary that highlights the main message of the teaching approach and add clarifications of the presented concepts.

Now that the structure of the teaching is established, the contents of the teaching method can be discussed. The aim of the teaching method is to improve the knowledge of the crowd workers in regards to software requirements so that they will be able to do the work presented to them more effectively and deliver higher quality. This requires a structured approach that works towards a homogeneous understanding of requirements, in order to properly distinguish between the different types of requirements to enable classification processes. We therefore introduce the requirements taxonomy defined by Glinz (2007), which is shown in Figure 14.

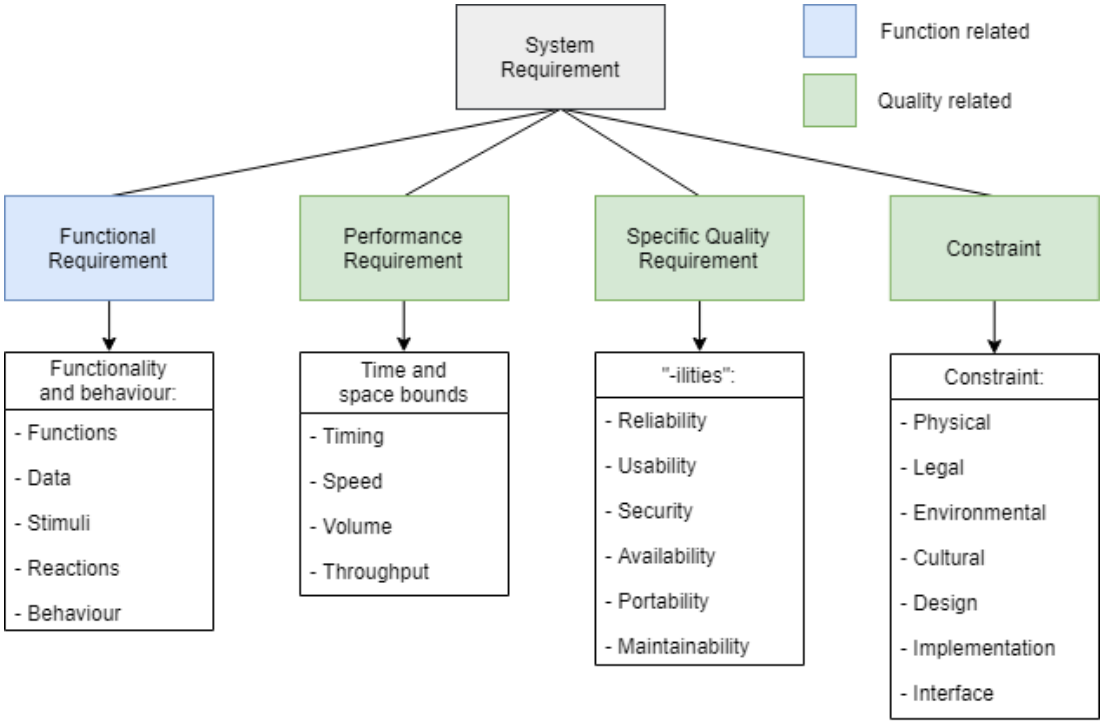


Figure 14: Requirements taxonomy by (Glinz, 2007).

The taxonomy starts off with establishing the term *system requirement* to indicate that the taxonomy concerns software related requirements to make explicit that it does not concern other types of requirements. It continues by distinguishing between functional requirements and three different types of quality aspects of a system to finally end up at the most detailed level of categories. This structure makes it suitable for classification purposes, as this taxonomy enables distinguishing between not only different types of requirements, but also because it introduces different levels of granularity, and therefore gives an indication about their level of detail. The classification of requirements can be guided by a set of questions introduced by Glinz (2007) that accompanied their proposed taxonomy. The classification process starts with a single question, with the right answer being the response that matches the most with the requirement that gets classified.

"Was this requirement stated because we need to specify..."

1. **Functional:** *"...some of the system's behavior, data, input, or reaction to input stimuli – regardless of the way how this is done?"*
2. **Performance** *"...restrictions about timing, processing or reaction speed, data volume, or throughput?"*
3. **Specific Quality:** *"...a specific quality that the system or a component shall have?"*
4. **Constraint:** *"...any other restriction about what the system shall do, how it shall do it, or any prescribed solution or solution element?"*

Both the taxonomy and the classification process can guide crowd workers with their contributions, as it introduces structure and guidelines for the classification process. However, as the quality framework of Allahbakhsh et al. (2013) indicates, the expertise of a worker is an important factor for the overall quality and value of the generated results. For the means of quality control and to test the effectiveness of the teaching approach, it will be important to test the knowledge of the contributing crowd members to get a sense of the general quality of their contributions. The exact way to test and measure the knowledge that the crowd contributors have about requirements will be explored in a later stage of the project.

2.5.4 Incentives

In earlier sections we distinguished between two different types of motivation, extrinsic and intrinsic. Fun and enjoyment are considered as the two leading intrinsic motivational factors prevalent

in online platforms (Hossain, 2012). Extrinsic motivators tend to focus on monetary rewards, which are very common in existing crowdsourcing practices and can be in the form of pay per task, pay for time and pay per annotation (Borromeo & Toyama, 2016). However, the reward amount generally only affects the amount of workers attracted and influences how fast they accomplish their task,increasing the amount does not necessarily increase outcome quality (Allahbakhsh et al., 2013) (Mason & Watts, 2010).

Crowdsourcing techniques that rely solely on intrinsic motivation also exist. Those techniques that rely on unpaid crowdsourcing must turn to other means to recruit volunteers such as users work in return for a service. This form of crowdsourcing is generally considered to be more challenging as they risk not having access to enough workers or being flooded by workers with malicious intent. On the other hand, unpaid crowdsourcing has a higher chance of acquiring highly motivated and skilled workers (Borromeo & Toyama, 2016)

A crowdsourcing technique that employs a mix of both extrinsic and intrinsic motivation should be able to gain access to the benefits of unpaid crowdsourcing and simultaneously reduce the risks that comes with it. To optimally motivate crowd workers, the to be developed technique should strive to employ a motivation policy that utilized both types of motivation.

Fun and enjoyment can be provided in multiple ways, but it is important to create a solution that fits the context, purpose and the goals of the crowdsourcing technique. Possible fun and enjoyment solutions could potentially be:

1. A sense of prestige in the form of reputation or other types of comparable scoring systems.
2. Recruiting crowd workers that have an emotional investment into specific software.
3. Providing praise and other types of feedback back to the crowd workers.
4. Show confirmation about the added value of the work performed by the crowd and how much it contributes towards the final goal.
5. A chance to win higher tier rewards or prizes for the contributor who provided the highest value.
6. Being open and transparent towards the crowd workers regarding progress.
7. Gaining more access to the developers and development processes of a particular software by creating elite group of crowd contributors based on their added value.
8. Providing earlier access to newer versions of software for crowd contributors.

2.5.5 Task dynamics

As previously discussed in section 2.2.2, crowdsourcing techniques appear to the most effective when the work is split into small micro-tasks with a predefined workflow, as this is the predominant form of crowdsourcing observed in practice (Valentine et al., 2017). Real-world problems however are often more complex tasks that require much more time and cognitive effort to solve (Malone, Malone, & Crowston, 1994), making them not as suitable for crowdsourcing solutions as micro-tasks. Splitting up the complex problem of requirements elicitation from online reviews into micro-tasks would require less time and effort from the crowd to complete their work and will likely result in higher quality results. However, this will have to be done in a structured way.

Kittur, Smus, Khamkar, and Kraut (2011) introduce the CrowdForge framework to coordinate the dependencies of aspects within complex task and their distribution into smaller tasks. The framework shown in Figure 15 aims to break down complex tasks systematically and dynamically into sequential or parallelizable tasks. Approaching this process structurally is important to recombine the generated results into a single final outcome of the complex task in later stages.

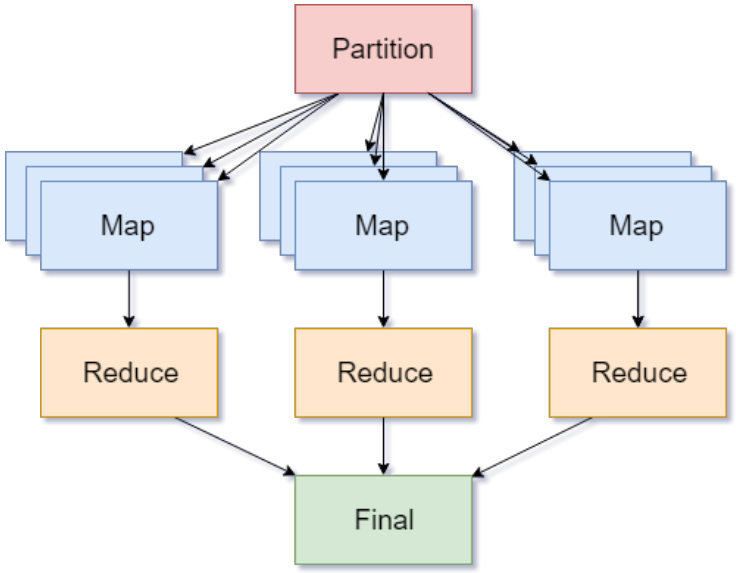


Figure 15: The CrowdForge complex task framework by (Kittur et al., 2011).

The CrowdForge framework introduced by Kittur et al. (2011) involves four elements that each aim to guide the splitting of a complex task into micro-tasks and to later correctly recombine the results generated by the crowd workers. These four layer are discussed below.

1. **Partition:** Breaking down a larger more complex task into smaller, distinctive subtasks.
2. **Map:** A simpler more specified task is processed by one or more workers.
3. **Reduce:** The results produced by the multiple workers are merged into a single output.
4. **Final:** The final outcome and the solution to the initial complex task.

While the framework indicates the partition phase as the first step, it does not necessarily mean that all partitions have to be completed before continuing The task designer is not required to know all possible subtasks initially, but can continue partitioning tasks into smaller tasks based on the reaction of the crowd workers. In other words, the exact division of labor and design of the subtasks can be based on the proceedings in practice. Kittur et al. (2011) mention this as an advantage that is novel and unique to human computation.

2.6 Main findings

We investigated the feasibility of crowdsourcing solutions in RE practices from three perspectives. By identifying the recognized opportunities and challenges from the areas of requirements engineering, crowdsourcing and natural language processing, a case is built for a more human-centred and crowdsourced RE solution. The earlier discussed opportunities for each three disciplines are summarized in the Figure below:

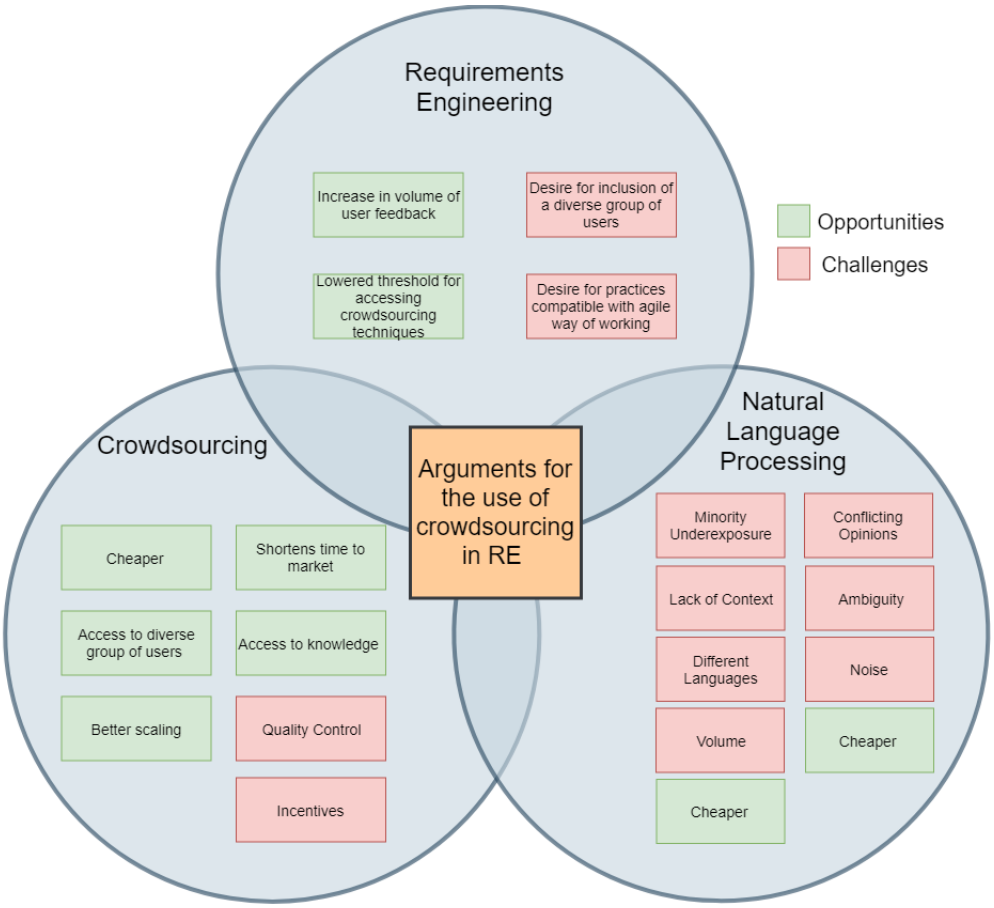


Figure 16: Summary of the arguments for the use of crowdsourcing in RE practices from three perspectives.

From the RE perspective, a shift from waterfall development approaches towards more agile ways of working has been identified. This is further substantiated by a representative of Deloitte during the interview shown in Section 2.5 which confirmed this observation occurring in practice. This agile way of working relies on quick development cycles, which requires RE practices that are able to incorporate user feedback into development processes more quickly than traditional approaches. Additionally, an influx in volume of available user feedback has been recognized caused by online review platforms such as mobile app stores. These platforms offer access to a new set of user feedback workable for RE practices, and can potentially satisfy the desire to include feedback from a more diverse group of users into development processes. The RE discipline has recognized the lowered threshold to gain access to crowdsourcing techniques and their potential application in practice.

Natural language processing practices already fulfill a role within RE practices and the processing of user generated feedback. Advances in the field show the potential to also provide the ability to automatically process user generated feedback from online reviews, but the unique nature of

the data introduce challenges that hinder their feasibility. The high amount of noise in the data, the presence of conflicting opinions and ambiguity, the use of different languages and overall lack of context complicate the analysis of the data through fully automated means. Questions arise whether fully automated means are the correct solution, as they cut out the human factor in the overall human-centred process.

Crowdsourcing solutions for the processing of user generated feedback present unique potential benefits and appear to be capable to deal with the nature of the data gathered from online reviews. Harnessing the power of the crowd may provide access to the required knowledge and processing power to properly interpret the user generated feedback and convert them into a set of workable requirements. Crowdsourcing techniques also appear to scale better, which makes them suitable to deal with the huge volume of provided feedback in a timely manner. Involving a crowd to generate these requirements and directing them to properly analyze and prioritize these requirements may be the correct approach from an cost and effort point of view, which in turn may make it economically feasible for companies as it can shorten the time-to-market for their software product.

Nevertheless, solutions that rely on crowdsourcing require proper management of the common pitfalls and challenges. Quality control and incentivizing the crowd to perform to the best of their ability remain the two biggest challenges. Reducing the negative effects that these challenges have on the results is one of the main objectives during the design phase of the project. To deal with these challenges and possible pitfalls, literature was explored that provide concrete frameworks or insights into applied practices that will guide the designing process.

We identified frameworks for quality control and mechanisms that can be implemented to influence the quality of the results. Additionally, we identified methods and approaches to involve and organize the crowd into crowdsourcing practices. Finally we explored ways to heighten the expertise of crowd workers, identified a taxonomy that enables the classification of requirements and explored methods that would guide the dynamics between the to be performed tasks and their partition into smaller subtasks. All aspects that have to be considered when designing a crowdsourcing solution are summarized in Table 4 on the next page.

General Approach	
Crowdsourcing Type	Directed / Collaborative / Passive
Task Structure	Simple / Complex / Creative
Crowd Characteristics	Marketing Branding / Productivity / Product/Service Innovation / Knowledge Capture
Crowd Involvement	
Contributor Preselection	Qualifiation-based / Context specific / Both
Peer Accessibility	None / View / Assess / Modify
Contribution Aggregation	Integrative / Selective
Remunerations	None / Fixed / Succes Based
Quality Control	
Worker Reputation	Explicitly / Implicitly
Worker Expertise	Credential based / Experience based
Task Definition	1) Description of nature and goals. 2) Completion criteria. 3) Eligibility criteria.
Task UI	Simple and intuitive, but advanced enough to improve performance.
Task Granularity	As simple as possible. Present required workflow to provide structure.
Task Compensation	Monetary to improve quantity of workers / Intrinsic to improve quality / Both
Training the Crowd	
Trigger	Make the workers familiar with the topic and objective of the task.
Learning Activity	Allow the workers to get used to the task by guiding them through the process.
Discussion	Discuss their performance of the learning activity and provide constructive feedback.
Summary	Summarize all of the involved concepts and their explanations and have them available for the crowd workers.
Incentives	
Extrinsic	Pay per task / Pay for Time / Pay per Contributions
Intrinsic	Sense of Prestige / Prizes / Preferential Treatment / Early Access
Task Dynamics	
Foundation	Explicitly define the work to be done and define the objectives and final state.
Partition	Break down the work to be done into logical, smaller and distinctive subtasks.
Map	Define each single subtasks in a way that is catered to the capabilities of the crowd workers.
Reduce	Define a way to combine the results from each individual subtask.
Final	Define the implications of the combined results from each individual subtask for the initial objective defined in the foundation.

Table 4: Summary of all elements and their possible choices involved in the development of a crowdsourcing technique.

In the next phase of this project, we will combine all of these aspects to guide the construction process of building a single, compact and robust crowdsourcing technique for the elicitation and classification of requirements from online user reviews.

3 Development of the Technique

All the literature presented in chapter two allows us to cautiously commence the design process of a technique for the elicitation of requirements from online reviews. The available literature covers the most prevalent and important aspects when working towards a crowdsourcing method. This chapter combines all these crowdsourcing aspects in an effort to inch towards a theorized approach as a starting point towards a concrete finalized method.

In its entirety, this chapter shows the creation of a crowdsourcing method for requirement elicitation and the progress towards that goal by discussing all arguments for different design decisions. It will weigh the different implementation possibilities based on feasibility, importance and practicality. Initial design decisions will be made on these characteristics to narrow down the possible forms of implementations to a single approach that is expected to be the most successful based on assumptions made from the literature. Additionally, available crowdsourcing platforms will be investigated, to get an overview of the real world possibilities and to factor in possible constraints of these platforms. The first iteration of the method will be constructed based on the combination of initial design decisions and the possibilities of available platforms.

This first iteration will be tested on a small scale to see how people react to the structure and the nature of the work and the data that they will be working with. Based on the feedback from these small scale tests, the method will be further developed towards a method that is the most likely to work optimally in a real, large scale test.

3.1 Design Decisions

As summarized in Table 4 in section 2.6, design decisions can be made on a multitude of different elements from different categories. The following sections will address these elements one by one per category and will work towards finding the optimal choices that will lead towards a first iteration of a crowdsourcing approach.

3.1.1 General Approach

Bigham et al. (2015) distinguished between three types of crowdsourcing; directed, collaborative and passive. Collaborative and passive crowdsourcing appear to be the most beneficial for the quality of the results, but are also significantly more demanding on areas of infrastructure or capabilities and motivation of the crowd. While collaborative and passive crowdsourcing have the potential to offer the most benefits, they require a large initial investment on aspects such as infrastructure and crowd recruitment to properly set up. In this stage of the project, there are too many uncertainties to warrant these large investments, so a **directed form** of crowdsourcing appears to be the appropriate way forward.

Crowdsourcing for the purpose of eliciting requirements from user reviews can be viewed as a complex task in its entirety, as it requires a high level of expertise from a specific field. This makes it costly to deploy in the form of crowdsourcing due to high amount of required incentives or remunerations and the high level of capabilities demanded of the crowd. Therefore, the crowdsourcing approach should **aim to to split up this work into multiple smaller microtasks** that are easier to grasp for the crowd. Downgrading the objectives of the task from complex problem solving, to extracting simple data or information from the crowd demand relatively small rewards and make the work small and more routine. However, to be effective it has to be deployed at a larger scale.

Tasks aimed to elicit requirements from user reviews best fit with the **product/service innovation** from the model of (Erickson et al., 2012). This means that this approach benefits the most from a large, diverse crowd who have distributed knowledge between them and that the crowd can be both recruited internally or externally to be maximally effective. This is beneficial, as it aligns with the notion that a directed form of crowdsourcing has to be deployed on a large scale to be effective.

The context of the work and the identified challenges from working with online user reviews require

the general structure of the work to be split in at least two parts. A first part should be dealing with the nature of the data, and the challenges introduce by spam and other irrelevant reviews. The second part should be revolving around working with the remaining useful data towards the classification of requirements. The exact structure will be discussed later.

3.1.2 Crowd Involvement

The four dimensions of crowd involvement by Geiger et al. (2011) show that there are multiple ways to involve a crowd in crowd work. The first dimension, the preselection of contributors, makes a distinction between a qualification based and a context-specific selection process.

Ideally, context specific selection will lead to the recruitment of a crowd that is already familiar with the topic of the work, and will therefore lead to higher quality results. Realistically, it will require a rigorous selection procedure that will severely limit the potential size of the crowd, as it would make it much harder to find people that are familiar with a specific topic such as requirements elicitation. As we have previously decided for a form of crowdsourcing that benefits the most from a large crowd, limiting ourselves to context specific selection will introduce too many implementation challenges and introduce the risk of harming the quality of the results. Therefore, a more open and less restrictive crowd selection policy should be considered. A **qualification based selection policy** would let everyone participate who can demonstrate that they posses a basic level of knowledge related to the task will likely be a fitting alternative. This way, it still has the possibility to attract people that are interested in the topic of the work, keep the potential to recruit a large scale crowd intact, but also keeping more control on the capabilities of the crowd.

The second dimension, peer accessibility, refers to the dynamics between the crowd workers themselves. It is possible to allow them to collaborate by allowing them to view, assess or modify contributions by other crowd workers, but this would introduce risks as well. Allowing workers to modify or assess the contribution may potentially lead to higher quality results but will also make the results more vulnerable to workers that operate in bad faith. Contributions may be altered incorrectly by one contributors to harm the reputation or the rewards for others for personal gain. Additionally, it would also make the work required much more complex as an additional element of reviewing other contributions would need to be introduced. Furthermore, allowing contributors to view other contributions may lead to the copying of answers or cause confusion when they are shown incorrect answers that can cloud their own judgments. At this point in time, no benefits could be identified that would warrant the additional risks introduced by solutions that allow collaboration compared to a more simple form that would only allow the workers to work separately from each other and allow **no interaction between them**.

The third dimension refers to the way that the results of the work will be aggregated when the job is completed. For selective aggregation, the outcome is based on the judgment of a contributor who is clearly and recognizably more capable in the task than its peers. Performance of individual crowd member can be reviewed based on their reputation, percentage of correct answers given or their performance on a potential test. However, as the current course of the method is a move towards simple directed crowdsourcing with no particularly strict selection policy, it is not guaranteed to attract these kinds of high performing contributors. Therefore, the alternative of integrative results aggregation becomes more attractive. **Integrative aggregation** pools all results together, only filtering out the answers that did not adhere to the predefined quality standards and bases the final answer on the majority vote. This makes a crowdsourcing method less dependant on the performance of a selective group of contributors and will benefit more from the collective knowledge of a large scale crowd.

The fourth and final dimension refers to the way that the crowd will be rewarded for their work. For a simple and directed form of crowdsourcing, it is unlikely that the crowd will participate voluntarily and not require any form of reward. This narrows down the options to fixed or success based remunerations. While either of these options does not appear to have any major distinctive benefits or downsides, one could argue that success based has the possibility to be demoralizing for a small, simple type of crowdsourcing that does not aim to recruit crowd workers on the long term. To not unnecessarily deter contributors from potentially participating, going with **fixed rewards** appears to be the less risky option. This decision can be reviewed and adjusted when necessary at

a later state.

3.1.3 Quality Control

The taxonomy of Allahbakhsh et al. (2013) showed us that both the crowd workers and the design of the tasks will affect the quality of the results. Furthermore, they distinguish between design time and run time quality control measures. As a simple and directed form of crowdsourcing has the highest risk to yield low quality results, the method should utilize as many quality control measures as possible from both of these dimensions.

The **design time quality control measures** can be taken into account mostly during the design phase of the crowdsourcing tasks. The four quality control elements that Allahbakhsh et al. (2013) define for task design are the definition, the user interface, the granularity of the tasks and the compensation for the crowd workers.

The **definition of the task** refers to the way that the work is presented to the crowd workers. The work will be clearly defined and include a context description that explains the nature and goals of the task, shows the completion criteria and the steps that the worker needs to perform to reach them and is transparent about the eligibility criteria to convey when a crowd worker can participate.

Secondly, the **task user interface** is recommended to be simple and intuitive, but also should be advanced enough to improve performance as a way to guide the workers through the work. Ideally, the task description is integrated in the part of the interface where the work is performed, to provide assistance when required. For this aspect, dependency lies on the possibilities provided by available crowdsourcing platforms. Due to the importance of the task user interface and the implications for the quality, it will be one of the selection criteria for the selection of a platform.

Subsequently the element of **task granularity** further confirms that the size of the tasks will affect the quality of the results. The method should strive to split the complex task into as small and modular tasks as possible. Each task should be stand on its own and should be able to be completed without knowledge of previous tasks. Complete separation will prevent the work from becoming unnecessarily confusing and will mitigate the risk for unnecessary mistakes. Lastly, specific instances of the required workflow should be provided to create structure and to guide the workers through the tasks.

The fourth element of task design refers to the **compensation of the tasks** and relates to the incentives for the crowd to complete the work. This will be discussed more in depth in one of the upcoming sections.

Additionally, the quality **control mechanisms for the run time element** will mostly affect the involvement of the crowd workers and the processing of the results. The method will require a way to indicate the expected capabilities of the crowd, as that will allow the restrictive preselection of contributors discussed earlier. According to Allahbakhsh et al. (2013), crowd capabilities can be derived from their **reputation based on their past performances**. From the two possible approaches to derive a reputation, the **explicit** form appears to be most appropriate way, if this can be provided by a platform In contrast, the implicit form requires an additional method to review the quality of the contributions by a crowd worker and to manually derive a reputation score from this. Due to the benefits and lower effort requirement of explicitly derived reputation, it will be one of the characteristics for the selection of a platform.

The final required quality control mechanism will revolve around the expertise of a crowd worker. To establish a base level of knowledge of the crowd workers, **a test will be deployed** that workers will have to pass in order to continue participating. A credential based mechanism utilizing documents or other official sources will be too unpractical and too restrictive for the topic of requirements engineering and is therefore not the favourable choice. Furthermore, the overall expertise of a worker in general can be increased by coaching, guiding and supporting the workers in doing their work, so the construction of a careful approach that introduces the workers to the topic and that provides guidance throughout will be the next focus.

3.1.4 Crowd Training

As crowd expertise is considered one of the main factors that will affect the quality of the results, the method will include an training aspect that will prepare the crowd for the tasks to be performed. To improve the capabilities of the crowd workers in an effective way, the training method will be based on the four aspects of active learning described by (Hazzan et al., 2015).

1. **Trigger:** Crowd workers will be given a description containing a general introduction about the purpose and the nature of the task and how the tasks can be completed. The description will be catered to audience, which means that the importance of requirements engineering and the value of feedback from user reviews will be quickly explained in layman's terms.
2. **Learning Activity:** Is where the crowd workers that were drawn in by the trigger will get their first exposure to the work itself. The learning activity will consist of three parts. A description will be provided that shows the steps that represent the desired workflow so that crowd workers can learn how they can complete the tasks. Secondly, guidelines and examples will be provided where the examples will show the correct answers and where the guidelines will show how the correct answer can be reached. Lastly, the final learning activity will be the the test that will allow or disallow people from contributing. It will consist of a couple of test questions that are in accordance with the guidelines and examples, so that crowd workers will be able to demonstrate that they understood those correctly.
3. **Discussion:** Will be where the answers of the test will be shown and discussed with the crowd worker. It will show whether they passed the test and are eligible to continue participating or failed to demonstrate the required level of knowledge. Regardless of passing, the incorrect answers will be shown and an additional description will be provided. This description will show how the answer relates to the guidelines and how they could have found the correct answer.
4. **Summary:** The entire set of descriptions, guidelines and examples will be summarized and made available to the crowd workers through the entirety of the work. The summary will cover all core aspects of the job to be done and will be structured in a way to quickly cover the possible immediate needs of the crowd workers.

As there is no real expectation to attract many intrinsically motivated crowd workers, it will be key to keep the training method as short and compact as possible to not take up too much of their valuable time. The effectiveness of the training method is related to the rewards that are offered for the work. The rewards should be sufficient enough to keep the workers interested long enough to invest their time in the training method.

3.1.5 Incentives

As discussed in section 3.1.2, rewarding the crowd worker in a fixed and consistent manner is assumed to be the less risky and therefore most safe approach. In crowdsourcing practices, extrinsic monetary rewards based on pay per task or pay for time are the most predominant ones and are shown to be effective in practice (Borromeo & Toyama, 2016). The monetary reward amount generally only affects the amount of workers attracted and influences how fast they accomplish their task, but increasing the amount does not necessarily increase outcome quality.

Intrinsic motivators have the potential to result in an outcome of higher quality, but are harder to implement. To not diminish the ability and potential to gather a large crowd, **extrinsic monetary rewards** appear to be the safe and logical choice. A simple and directed form of crowdsourcing that is aimed at problem solving benefits the most from a large crowd, and we should therefore strive to not inhibit this from occurring. The chance of yielding lower quality results can be mitigated by other (previously discussed) quality controls and a more comprehensive analysis method. The choice for simple monetary rewards can be reconsidered when the method shows signs of struggling to reach sufficient respondents or when the jobs takes a lot of time to complete.

Tests will be performed to get an indication of the time required to complete the task and to set a baseline for the monetary reward per task. Horton and Chilton (2010) present a model that aims to approach the minimum wage that workers will accept to perform the work, but requires detailed demographic information about the participants. As this information is nonexistent for

the context of this project, the solution will be setting the reward at approximately minimum hourly wage and make adjustments along the way based on responses.

3.1.6 Task Dynamics

The previously introduced categories of crowdsourcing tasks defined by Schenk and Guittard (2011) show the characteristics, benefits and downsides of the three task types. The process of eliciting requirements from user reviews and classifying them into their respective categories can be interpreted as a problem solving activity, and therefore falls into the category of a complex task. According to the task types of Schenk and Guittard (2011), this will require a high level of expertise of the crowd, make it costly to incentivize the crowd workers and has the possibility that the right solution will not be found. In contrast, simple tasks require a low level of expertise, are generally not costly and have relatively low problems in regard to incentivizing the crowd workers.

Simple tasks are the most predominant form of task in crowdsourcing practices for the reasons listed above. Simple micro-task workflows are for these reasons the dominant crowdsourcing structure in practice (Valentine et al., 2017), but require highly controlled predefined workflows to manage paid, non-expert workers toward expert-level results (Retelny et al., 2014). Ideally, complex tasks should be split up into microtasks to reap the benefits that small and simple tasks offer, but this requires an approach that structurally transforms complex tasks into a set of more simple tasks.

It is possible to apply the steps of the CrowdForge framework by Kittur et al. (2011) discussed in section 2.4.5 to the classification process of the requirements taxonomy by Glinz (2007). The partition can be based on the different levels of granularity in the taxonomy to **separate the classification process**. The different map tasks can be focused around filtering user reviews based on usefulness, identifying user needs from user reviews and classifying them as functional or quality aspect and lastly to classify them into their respective category. Following the framework, having separate groups of working assigned to these tasks and combining the results from each of these individual mappings allows the creation of a single final output, which will be final requirements. Using the CrowdForge framework the guide the process of splitting a complex task such as requirements elicitation into smaller, more modular tasks, appears to be the right move forward.

3.1.7 Conclusion

A simple and directed form of crowdsourcing appears to be the most feasible option out of all possible approaches. We will aim to split up the requirements elicitation process in as many simple and standalone tasks as possible and adjusted to the capabilities of the crowd. Crowd workers will be required to pass an eligibility test and will work separated from each other.

Direct and consistent monetary rewards will be offered as compensation for the work. Other types of rewards are assumed to be detrimental for the ability to gather a large crowd of contributors. To compensate for the likely outcome of yielding lower quality results compared to possible alternatives, multiple quality control mechanism will have to be considered. The tasks will be designed to be small, completely standalone and will not be made to be too demanding on specific and complex knowledge. An elaborate learning method will be developed and deployed alongside the work that will quickly provide the workers with the knowledge to complete their tasks to the best of their abilities. Lastly, the tasks will be presented through an intuitive design that will cooperate with the required workflow as much as possible. The workflow itself will use the requirements classification taxonomy by Glinz (2007) as a starting point.

All discussed design decisions are summarized in Table 5 shown in the next page.

General Approach	
Crowdsourcing Type	Directed
Task Structure	Simple
Crowd Characteristics	Product/Service Innovation
Crowd Involvement	
Contributor Preselection	Qualifiation-based
Peer Accessibility	None
Contribution Aggregation	Integrative
Remunerations	Fixed
Quality Control	
Worker Reputation	Explicitly
Worker Expertise	Experience based
Task Definition	1) Description of nature and goals. 2) Completion criteria. 3) Eligibility criteria.
Task UI	Simple and intuitive, but advanced enough to improve performance.
Task Granularity	As simple as possible. Present required workflow to provide structure.
Task Compensation	Monetary to improve quantity of workers
Training the Crowd	
Trigger	Make the workers familiar with the topic and objective of the task.
Learning Activity	Allow the workers to get used to the task by guiding them through the process.
Discussion	Discuss their performance of the learning activity and provide constructive feedback.
Summary	Summarize all of the involved concepts and their explanations and have them available for the crowd workers.
Incentives	
Extrinsic	Pay per Contributions
Intrinsic	-
Task Dynamics	
Foundation	The processing of user generated feedback to elicit useful information for software developers.
Partition	Separating the requirements elicitation process into multiple steps that entail dealing with the bad nature of the data and the classification process.
Map	Distinguishing between useful and useless reviews. Classifying useful reviews into different categories.
Reduce	Aggregating the results generated by all crowd workers for each processed reviews.
Final	Useless results will be filtered out and removed from the dataset. Useful reviews will be classified in several different requirements categories.

Table 5: Summary of the current main design decisions for the to be constructed technique.

3.2 Platform Investigation

Crowdsourcing in itself is not a particularly new concept, as evidenced by the prevalence of available crowdsourcing platforms on the world wide web. Earlier we already identified Amazon Mechanical Turk as a potential platform for future crowdsourcing practices, but without having knowledge about other options to deploy crowdsourced work. Therefore, this section investigates those existing crowdsourcing platforms that appear to have the necessary infrastructure to host a series of tasks for the purpose of requirements elicitation.

In this section, we review several crowdsourcing platforms to identify the platform that corresponds the most with the design decisions described in section 3.1. During these reviews, comparisons will be made on a couple of key aspects such as accessibility and customizability to ensure that the platform is a feasible option for a large scale real-world testing. Furthermore, key design decisions such as task user interface and quality control mechanisms will be considered, as they were deemed to be the most influential design factors on the quality of the results.

3.2.1 Amazon Mechanical Turk

Perhaps the most well known and most studied platform, Amazon Mechanical Turk has been at the forefront of early crowdsourcing practices. Studies going back to 2010 accumulated thousands of references and already praised the platform and preached its viability for crowdsourcing purposes (Paolacci, Chandler, & Ipeirotis, 2010). Due to its reputation, Amazon Mechanical Turk can be viewed as the largest provider that facilitates exchanges between work requesters and crowd workers. Therefore, availability of crowd workers therefore does not appear to be a limitation when deploying crowdsourcing tasks on this platform.

While previously only accessible by work requesters that were in possession of a bank account linked to American banks, funds can be added to accounts worldwide with the use of credit cards as well. Plenty of options could be identified for setting up a crowdsourcing tasks, but no premade template for data categorization could be identified. Most options are catered to surveys, image recognition and linguistic and language analysis finding no real fit with a classification job required for the elicitation of requirements. Furthermore, tasks have the be custom made and structured using HTML, CSS and Javascript by the work requester. While limitations could be imposed on participants as a means of quality control, no advanced mechanisms could be identified apart from worker reputation.

3.2.2 Figure Eight

Formerly known as CrowdFlower, Figure Eight has been around since the year 2007 but underwent a lot of changes before becoming an open platform for crowdsourcing work. Starting as an experiment on Amazon Mechanical Turk itself, it was founded as a platform where human effort could be utilized for the training of machine learning techniques. Currently, Figure Eight functions as an aggregation of multiple other channels that provide crowd workers, in addition to their own contributor portal.

Figure Eight is an easily accessible platform with different access portals for both the workers and the work requesters. The work requester platform provides extensive templates for different categories of jobs and with data categorization as one of the main ones. An online editor tool is available that allows the work requester to quickly customize the job itself, its workflow, the accompanying description and the organization of the results. Creating an account is free, but trial accounts are limited to 1,000 rows of data and the platform itself takes up 20% of the costs involved when launching a job. Furthermore, a multitude of quality controls are available that allows the work requester to set up test questions that crowd workers have to pass before being allowed to contribute. The platform has an in-built reputation system that gives an indication of the capabilities of the crowd workers based on past performances. Lastly, options are offered to exclude crowd workers from specific countries or channels from participating.

3.2.3 Other

In addition to the previously mentioned platforms, multiple other available platforms were also explored and treated as candidate platforms. However, they could be quickly discarded as candidate platforms due to the focus of the platform or the limited accessibility options. A platform as RapidWorkers for example is a crowdsourcing service with 100,000 workers, but is only aimed at very small promotional tasks such as voting for Youtube videos or buying Twitter followers. Furthermore, Samasource also appeared to be a suitable platform for crowd work, but turned out the be not accessible for third parties due to their focus on providing micro tasks to people in development countries. Other available platforms such as InnoCentive, Zooppa and Remesh focused only on complex problem solving or other types of work that required a high degree of creativity. Due to the misalignment between the purpose of these platforms and the intentions of this project, we decided to discard them and not analyze them further. Lastly, platforms such as MicroWorkers or ClickWorker that did not have this specific focus on one type of job, could be discarded on the basis on the perceived ineffectiveness of task design and user interface.

3.2.4 Conclusion

The purpose of this section is to identify a platform that fits best with the design decisions in section 3.1, is accessible and customizable enough to deploy crowdsourcing tasks for the elicitation of requirements. Rather quickly, it became apparent that Amazon Mechanical Turk and Figure Eight are most serious contenders, as they are the most advanced, accredited and well known platforms. Furthermore, Amazon Mechanical Turk and Figure Eight both offer the broadest options, making them suitable for multiple different kinds of crowdsourcing approaches, compared to smaller platforms that only focus one a single approach or purpose.

Studies have been conducted where Amazon Mechanical Turk and Figure Eight (CrowdFlower) are compared based on their options and results for identical tasks (Finin et al., 2010). The most important finding stated by Finin et al. (2010) was that both platforms are flexible, easy to use, capable of producing usable data and are very cost effective. Nevertheless, the authors gave a slight edge to Figure Eight due to their extra features and interface options. These observations are in line with the observations from this review and therefore make the Figure Eight platform the most suitable candidate for future tests in the context of this project.

3.3 First Iteration

With the initial design decisions in place and the crowdsourcing platform selected, the focus can be shifted towards the first iteration of the method. The method will have to be able to deal with different challenges introduced by the nature of user reviews and the difficulty of the classification process. Therefore, the first iteration will consist of at least two different parts. The first part will deal with the nature of the data of online reviews and will aim to filter out all irrelevant and spam reviews. The second part of the method will be based on the RE taxonomy of Glinz (2007), as a foundation for the requirements classification process.

To deal with the noise within the data, the first part will be split into two separate phases. It will consist of one phase dealing with spam reviews and the other focused on filtering out irrelevant results. Here, spam refers to ineligibly written reviews that do not convey any kind of well intended message. On the other hand, irrelevant reviews are when a certain message is conveyed but it serves no purpose for the developers of the software. For the first phase, entire unprocessed reviews will be used as input and will have as output a set of reviews with the spam removed. This set of reviews without spam will be the input for the second phase, which will filter out all irrelevant reviews.

The second part of the method will focus on the classification of requirements in accordance with the requirements taxonomy by Glinz (2007). The classification process will be divided into three separate phases. The first phase from the categorization process will use the output from the second phase, which is the set of reviews that are left when all spam and irrelevant reviews are filtered out. This phase will focus on separating functional requests from the others, as this is the first step of the classification taxonomy by Glinz (2007). The succeeding phase will take the

reviews that were not classified as functional requests and aim to classify them as performance related, specific quality related or a constraint. Lastly, the final phase does not necessarily has to be completed in one single tasks, It uses the input from the four main categories and aims to classify the reviews belonging to each individual category into the multiple subcategories that Glinz (2007) proposes.

Based on all discussed design principles, all different phases required for a crowdsourcing approach for the elicitation and classification of requirements from online user reviews looks like the following, as depicted in Figure 17.

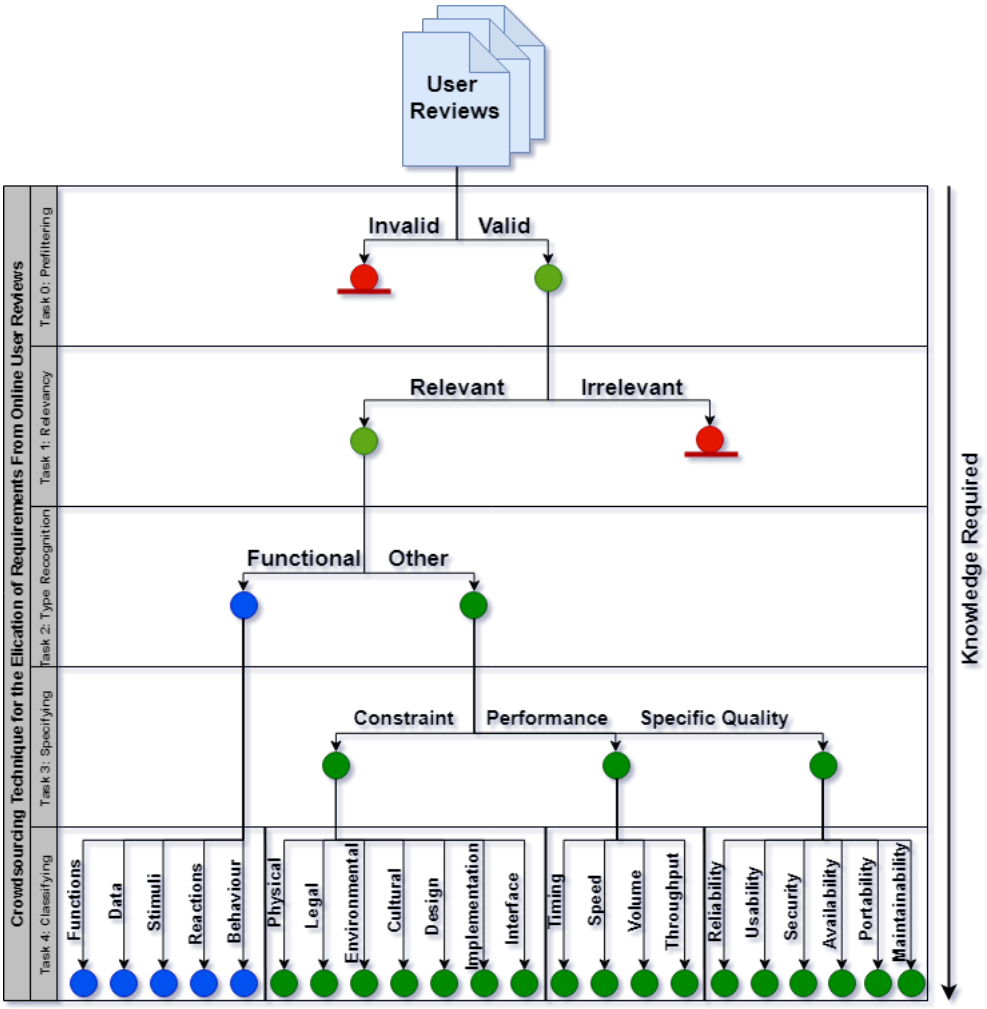


Figure 17: Visualization of the first iteration of the method.

1. **Prefiltering:** Filtering out the reviews that are unusable due to spam or incompleteness.
2. **Relevance:** Indicating whether they discuss a possible requirement or a specific need of a user by tagging the relevant part of the user review.
3. **Type Recognition:** Recognizing whether the requirement or conveyed user need belongs to the functional or quality aspect category.
4. **Specifying:** Classifying the needs that indicate quality aspects of a system and provide insight into the performance, specific quality or constraints of a system.
5. **Classifying:** Classifying the needs into their respective categories with the lowest level of detail.

Using this first iteration of possible method, further design processes can commence for further refinement. Before being ready for deployment in an large scale crowdsourcing test, several small

tests will have to be performed to see how people react to this approach and to further shape the method based on their performance and feedback.

3.4 Internal testing

While most of the design decisions for the first iteration were made based on the discussed literature, some assumptions had to be made on areas that the literature did not cover or where real world experience was lacking or non-existent. Before being ready for large scale testing, the first iteration will be subjected to some smaller tests to see how people react to the current approach. These smaller test will focus on discovering how the major design decisions hold up in practice and to see where it can be improved. Based on the feedback from these smaller tests, the method will be shaped and refined further to filter out any major errors and wrong assumptions and to check the feasibility of the approach.

The tests will be aimed at the two main objectives of the approach, filtering out user reviews that are not useful and the classification of useful reviews into different RE categories. The exact objectives of the two internal tests are discussed below.

1. **First internal test:** Testing the ability of the participants to distinguish useful reviews from invalid and/or irrelevant reviews. The results from this test will be compared against two different benchmarks, to both check the ability of participants to recognize illegitimate reviews (spam) or to recognize reviews that are legitimate but hold no valuable information.

   (a) **First Benchmark:** The ability to identify an arbitrary number of spam reviews from a large set of diverse user reviews.

   (b) **Second Benchmark:** The ability to identify an arbitrary number of irrelevant reviews, in addition to the spam reviews.

   A single description shall be provided to the participants, which is catered to the identification of spam reviews only. The description will be constructed in accordance with the crowd training principles discussed in section 3.1.4. The results will be interpreted for both benchmarks in order to compare the capabilities of participants to recognize both spam and other irrelevant reviews. This comparison will show how the participants interpret the description and how it influences their decision making. Potential adjustments or changes can be made based on the participant performance of either benchmark.
2. **Second internal test:** The second test will have a similar setup to the first internal test, but will aim to test the ability of participants to classify helpful reviews into four main categories. The results will be interpreted for each individual category to get an overview of the capabilities of the participants to recognize each category. Adjustments will be based on the proportion of wrongly classified results between different categories.

These internal tests will be conducted using Google Forms, as a mock-up version of crowd tasks on existing platforms. The setup will aim to mimic the interface and workflow of existing crowdsourcing platforms to get as close to a real world scenario as possible. In addition to the answers, the time spent to complete the entire task will also be recorded to get an indication about the time required to complete tasks in this format. Lastly, the test will be distributed to college educated individuals, including participants both internally in the department of Information and Computing Science at Utrecht University and externally to include people that are not familiar with the RE domain.

3.4.1 First Internal Test

For the first internal test, 50 user reviews were manually selected and consisted of 5 reviews that were clearly spam, 10 reviews that were not spam but had no useful or relevant information for developers and 35 reviews that were constructive and helpful. This distribution is not necessarily representative for the entire dataset, but it will aid in discovering how participants react to these different kinds of reviews.

The task description for this test only distinguished between valid and invalid reviews, where invalid refers to the reviews that could be interpreted as spam. The valid reviews were described

as any kind of review that appeared to be written genuinely and conveyed any kind of content or opinions related to an app. The entire description and its steps, guidelines and examples that was provided to the participants are shown below.

Description
In this job, you will be presented with text from user reviews from mobile app stores such as the Google Play Store or the Apple App Store. User reviews can hold important information for developers, as feedback from users is important for the evolution of their product. However, before developers gain access to the valuable information that user reviews offer, they require processing.

The goal of this job is to filter out the spam and to remove illegitimate reviews. This entails classifying individual reviews as either valid or invalid, based on a set of guidelines.

Steps

1. Read each user review carefully.
2. Determine whether the review is valid or invalid based on the guidelines listed below.
3. Mark the reviews valid or invalid.

Continue to the next review and repeat steps 1-3.

Guidelines
For each of the feedback sentences, carefully determine the relevant feedback category.

- **Invalid Reviews**
  - Contain spam or other unrequested or unwanted messages in bulk.
  - Have content that is unrelated to the app and its functions.
  - Are unreadable and/or contain a lot of spelling errors.
- **Valid Reviews**
  - Appear to be written genuinely and don't contain jokes or absurd statements and claims.
  - Mention aspects of the applications regarding functions, features, bugs and performance issues.
  - Present constructive criticism and/or opinions from users.

Continues on next page.

Examples

- **Valid Reviews**
  - *"It would be nice to stream my library to chromecast then I would rate 5 star"*
  - *"Very user friendly"*
  - *"Buggy and unreliable. Does not work often. Signs me out regularly. Won't download movies onto my iPad. Disappointing."*
  - *"Ok, but design could be better, too dark"*
  - *"Fun but so data hungry. Why does it need so much data???"*
- **Invalid Reviews**
  - *"WRSET APP OF THE CENTUREY"*
  - *"Pilemdmkdmdmdmfmmfmmf"*
  - *"1st!"*
  - *"Impressions 0_0"*
  - *"Goodwood whaaaaaaaqueeeeeeeem"*

Using this description, the first internal test was conducted by using Google Forms. A total of 9 participants contributed to this internal test where each of them classified 50 reviews as valid or invalid. As intended during the design, the description only explicitly mentions spam and other illegitimate reviews. Table 6 shows all 50 reviews together with the percentage of participants that classified them as invalid.

Review	Correct %	Review	Correct %	Review	Correct %	Review	Correct %	Review	Correct %
1	100%	11	100%	21	89%	31	78%	41	66%
2	100%	12	78%	22	89%	32	78%	42	66%
3	100%	13	89%	23	100%	33	89%	43	89%
4	100%	14	78%	24	89%	34	100%	44	45%
5	100%	15	44%	25	100%	35	89%	45	55%
6	66%	16	89%	26	66%	36	78%	46	89%
7	78%	17	45%	27	89%	37	78%	47	100%
8	78%	18	89%	28	33%	38	100%	48	100%
9	100%	19	100%	29	45%	39	78%	49	100%
10	100%	20	78%	30	78%	40	100%	50	100%

Table 6: Individual reviews and the percentage of participants that marked them correctly.

Reviews 1-5 were the reviews that should have been classified as invalid as those were the reviews fitting the description of spam and are marked as red. Reviews 6-15 did not fit the definition of spam, but did not contain relevant information for developers and are marked as orange. Analyzing the results in Table 6 it becomes evident that even though all participants were presented the reviews in a randomized order, they all correctly marked the 5 spam reviews as invalid. However, a substantial amount of wrong answers can also be observed, where irrelevant reviews were also marked as invalid by the participants even though that was not the objective described in the description.

Participant	Duration (Sec)	Nr. Valids	Nr. Invalids	TP (5)	TN (45)	FP	FN	Precision	Recall
1	332	27	23	5	27	18	0	0,22	1
2	218	33	17	5	33	12	0	0,29	1
3	697	30	20	5	30	15	0	0,25	1
4	1674	29	21	5	29	16	0	0,24	1
5	774	33	17	5	33	12	0	0,29	1
6	494	34	16	5	34	11	0	0,31	1
7	251	18	32	5	18	27	0	0,16	1
8	171	39	11	5	39	6	0	0,45	1
9	414	21	19	5	31	14	0	0,26	1
Average:	**558 (9,3 Minutes)**	**29,3**	**19,6**	**5**	**30,4**	**14,6**	**0**	**0,28**	**1**

Table 7: Results of the first internal test for the first benchmark.

This effect is further emphasized by analyzing the precision score of the participants, as shown in Table 7. Even when they perfectly identified all five spam reviews as invalid (as indicated by the perfect recall score of 1), the average precision score of 0.28 indicates that they marked reviews as invalid nearly four times more than intended.

Participant	Duration (Sec)	Nr. Valids	Nr. Not Valids	TP (15)	TN (35)	FP	FN	Precision	Recall
1	332	27	23	14	24	11	1	0,56	0,93
2	218	33	17	14	32	3	1	0,82	0,93
3	697	30	20	10	25	10	5	0,50	0,67
4	1674	29	21	15	29	6	0	0,71	1,00
5	774	33	17	14	32	3	1	0,82	0,93
6	494	34	16	15	34	1	0	0,94	1,00
7	251	18	32	13	16	19	2	0,41	0,87
8	171	39	11	8	32	3	7	0,73	0,53
9	414	21	19	15	31	4	0	0,79	1,00
Average:	558 (9,3 Minutes)	29,3	19,6	13,1	28,3	6,7	1,9	0,70	0,87

Table 8: Results of the first internal test for the second benchmark.

When moving the objective of the task to the second benchmark, the results tell an entirely different story. When the irrelevant reviews are also included in the set of correct answers, a large shift can be observed in the precision of the participants. Instead of only marking 28% of the reviews correctly as invalid, now 70% of the 15 spam or irrelevant reviews were correctly marked as invalid. As the description only specifically mentions spam reviews, we can conclude that participants are clearly able to identify both spam and irrelevant reviews at the same time. However, the overall recall decreased a little, meaning not all spam and irrelevant reviews are filtered out of the set of 50 reviews. This can likely be accredited to the lacking description that does not cover the relevancy of reviews whatsoever.

Due to the different results between the two benchmarks, we are inclined to conclude that participants are capable of identifying both spam and irrelevant reviews in the same task. Combining the first and second phase of the method and catering the description to both spam and irrelevant reviews appears to be the most logical way forward.

Furthermore, Table 6 indicates that 4 reviews were incorrectly marked as invalid by the majority of the participants. Specifically these four reviews give an indication where the current description is lacking.

Number	Review
17	Real good game I really like this game but can you can you please fix the fidgity controls because at the start instead of pac-man going straight up he goes up and then he starts going side to aside and when I swipe up he won't go up pleeeeeeeeeeeeeeease fix and I will rate five stars :):):):):):):):):)
28	Slow unstable 5s
29	This viber is good but i find it a little slow but its great
44	Bring back old interface

Table 9: Reviews that were incorrectly marked as invalid by the participants.

Reviews 28 and 29 indicate that the relevancy of performance related feedback should be covered better in the description. For later stages of the method, reviews mentioning the speed or reliability of an application are very much relevant. Additionally, review 17 does hold relevant information but is assumed to be filtered out due to the mention of hard to read reviews as invalid in the description, as well as one of the examples mentioning the use of smileys. To decrease the chance of participants seeing similar reviews as invalid, it may be the best approach to remove these parts from the description. Finally, the average time of 9,3 minutes that the participants required to classify 50 reviews gives an useful approximation for the decision on sensible monetary rewards in later stages.

3.4.2 Second Internal Test

The second internal test was carried out with a similar setup as the first internal test. A total of 50 reviews were hand picked and set up in a classification task. Participants were offered five different

categories; feature request, reliability feedback, performance feedback, quality feedback and none of the above. The selected reviews that were used in the test were selected based on overall clarity and their degree of ambiguity. Reviews that could be interpreted in multiple ways or that were technically eligible to be categorized in multiple categories were avoided during the selection. All this this resulted in a distribution of 9 feature requests, 10 for reliability feedback, 9 performances related reviews, 12 quality related reviews and 10 that belonged in none of the other categories.

Similar to the first internal test, a description was prepared giving a brief introduction, showed the required workflow, provided guidelines and showed examples of correctly classified reviews. The information provided to the participants alongside their tasks is shown below.

Description
In this job, you will be presented with fragments of user reviews from mobile app stores such as the Google Play Store or the Apple App Store. User feedback is important for the evolution and improvement of apps and can come in many different forms.

The goal of this job is **to identify what type of feedback the review mentions**. This entails classifying individual reviews as either being a **feature request, reliability, performance, quality** related or **none** at all based on the guidelines below. If the feedback sentence fits **multiple categories** according to the guidelines, please choose the **predominant** one.

Steps

1. Read each feedback sentence carefully.
2. Determine whether the feedback sentence matches with one of the categories based on the guidelines and examples listed below.
3. Mark the feedback sentence as **Feature Request, Reliability Feedback, Performance Feedback, Quality Feedback** or **None** of the above

Continue to the next review and repeat steps 1-3.

Guidelines
For each of the feedback sentences, carefully determine the relevant feedback category.

- **Feature Request:**
  - Expresses a desire for functions to be implemented in the future.
  - Mention what features or functions are missing according to the user.
- **Reliability Feedback:**
  - Refers to crashes or other instances when the application froze or did not work at all.
  - Mention bugs that completely prevented users from interacting with the application.
- **Performance Feedback:**
  - Refers to how quickly the user can interact with the application interface.
  - Mentions how quick or slow the application performs after the user performs specific actions.
- **Quality Feedback:**
  - Explicitly mention what the users liked or disliked about the application (not a generic "I love this app").
  - Often refers to how easy to use the application is perceived by the user.
- **None of the above:**
  - Do not appear to mention information relevant to an application of any kind.
  - Mention other aspects about an application that clearly does not fit with the other previous categories.

Continues on next page.

Examples

- **Feature Request**
  - *"I just wish I could turn off the emojis because I don't like them at all."*
  - *" Add video calling and you might have a contender for Skype."*
  - *"It's nice but lacks the download feature that I have on the app for my android phone"*
- **Reliability Feedback**
  - *"Newest version crashes when opening"*
  - *"Buggy and unreliable, does not work often and signs me out regularly."*
  - *"Still not working, keeps crashing after login details are entered."*
- **Performance Feedback**
  - *"Good app but it is slow and lags sometimes."*
  - *"The most recent update improved the speed."*
  - *"Stalls every time you write a tweet."*
- **Quality Feedback**
  - *"Both the texting and calling functionalities are easy to use."*
  - *"It is smart, quick, and compact.*
  - *"Very customizable and clean lay out."*
- **None of the above**
  - *"My kids love it. Thanks"*
  - *"Not even worth the download time."*
  - *"This is by far the worst app ever. I hate it so much."*

This description and the reviews again were set up in Google Forms and distributed to 11 willing participants. The distribution of their answers is shown in Figure 10.

Participant	Duration (Sec)	None of the above (10)	Feature Request (9)	Reliability Feedback (10)	Performance Feedback (9)	Quality Feedback (12)
1	475	10	7	10	7	16
2	1356	12	8	11	9	10
3	522	12	9	9	11	9
4	451	9	10	7	13	11
5	551	6	11	10	7	16
6	1657	7	9	10	6	18
7	776	6	12	11	10	11
8	399	14	7	5	6	18
9	687	14	10	11	8	7
10	616	5	11	10	6	18
11	679	7	10	11	8	14
Average:	**743 (12,4 Minutes)**	**9,3**	**9,5**	**9,5**	**8,3**	**13,5**

Table 10: Distribution of the answers given by the participants of the second internal test.

The participants required on average around 12,4 minutes to complete the classification process for the 50 reviews. Out of the available categories, the quality feedback category was chosen the most with around 13,5 times on average per participant. The performance category was chosen the least by the participant with only being selected 8,3 times out of 50 reviews on average. However, the correctness of the answers will have to be involved to get a more comprehensive image of the effectiveness of the classification process.

Category	Nr. Selected	Actual Nr.	TP	TN	FP	FN	Precision	Recall
None	102	110	69	407	33	41	0,68	0,63
Feature	104	99	91	438	13	8	0,88	0,92
Reliability	105	110	85	420	20	25	0,81	0,77
Performance	91	99	61	421	30	38	0,67	0,62
Quality	148	132	92	361	56	40	0,62	0,70

Table 11: Distribution of the selected categories and their precision and recall values.

As depicted in Figure 11, the aggregated results for all participants show that all categories were selected for a certain amount of times that does not differ too much from the actual classification. The quality category was chosen by far the most and more than intended, with being selected a total of 148 times out of the possible 132. The values for the other categories have much smaller deviations. However, the precision and recall values show us that a substantial amount of errors were made in the quality, performance and the none category. On average, only around two thirds of the reviews that were classified as such were correct. On the other hand however, the feature and reliability category were correctly selected between 80 and 90 percent of the time.

		Participant Judgments				
	Category	**None**	**Feature**	**Reliability**	**Performance**	**Quality**
Actual	**None**	69	3	3	5	30
	Feature	1	91	1	0	6
	Reliability	6	2	85	14	3
	Performance	4	1	16	61	17
	Quality	22	7	0	11	92

Figure 18: Confusion Matrix comparing the answers of the participants with the actual correct answers.

The confusion matrix in Figure 18 allows us to identify the areas where errors were most common. In the instance of this internal test, three particular areas with disparities between two different categories can be identified. The two categories that were interchanged the most were none of the above and quality. Participants selected none of the above 22 times when it should have been quality. Vice versa, the participants selected quality 30 times when it should have been none of the above. Although to a lesser degree, similar effects can be observed between the reliability and performance categories and the performance and quality categories. These disparities indicate that the task description in its current form is not effective enough in guiding workers towards the correct decision.

3.4.3 Design Implications

The most valuable conclusion that could be drawn from the first internal test is that participants were more than capable of identifying both spam and irrelevant reviews at the same time. Even with the description only catered to the identification of spam reviews, participants were way more accurate in their judgments when irrelevant reviews were included as correct answers. On the basis of this notion, the decision was made to combine the first two phases of the method into one single phase. Spam and irrelevant reviews will be combined under a category called *useless*. The objective of the task will shift toward distinguishing *useless* reviews from reviews that hold *helpful* information. Finally, other possible improvements could be identified due to the reviews that were incorrectly marked as useless. In response, the relevance of performance related feedback will be emphasized more. Lastly, the example that used a invalid review containing smileys will be removed to not suggest that all reviews containing smileys are invalid.

Another important observation was made independently of the results of the participants during the conduction of the first internal test. Online reviews do not always have a small character limit, which makes it possible that very large reviews of multiple sentences will also be included in this method. The possibility exists that each individual sentence will mention a unique requirement or that only one sentence out of many contains helpful information. To deal with reviews of multiple sentences, an additional phase will be added. During this phase, reviews that were marked as helpful will be split per sentence using an automatic text processor. These individual sentences will be viewed as fragments of user reviews and will be exposed to a nearly identical classification process as in phase 1. The output of this second phase will be helpful fragments of user reviews that contain requirements and will be ready for further refinement.

The design of the classification part of the method used in the second internal test already deviated quite a bit from the first iteration. The categorization part was simplified due to the realization

that the classification process from the taxonomy clashed with one of the core principles of crowdsourcing, stating to keep tasks as simple as possible. Thus, instead of using all the exact terms and categories of the taxonomy of Glinz (2007), the number of categories was reduced to five. Additionally, more general terms were used that still encapsulated the core categories of the taxonomy, but made them more recognizable for the crowd workers. Nevertheless, the second internal test using this setup gave plenty of indications of where the current approach could be improved and refined.

Due to the overlap between the reliability and performance category, reliability was renamed to stability. This term is expected to be more recognizable and also better conveys the intended meaning of the category. The stability category will cover the availability and fault tolerance of a system and should refer to reviews that convey that some sort of crash happened. Secondly, an element of resource utilization was added to the performance category to make it more distinctive from the quality category. The performance category will now not only cover speed or slowness of an application, but also battery, memory or internet usage behaviour. Furthermore, an example referring to privacy and password recovery was added under the quality category. This should make it more clear that reviews with this topic are indeed relevant and should not be classified under the none category. Lastly, an additional conceptual phase was added due to the derivation from the taxonomy of Glinz (2007). The simplification of the approach to only have 5 different categories for classification could be perceived as not advanced enough, the method therefore requires optional deeper layers for further refinement of the classification when necessary. Distinguishing function requests and interoperability requests from the previous feature category or refining the quality aspects of a systems into more specific categories could be two possible uses for these extra phases.

3.5 Final Design

The design implications both derived from the construction and the conduction of the internal test unearthed plenty of aspects of where the method could be improved. While both major and minor design implications could be discovered, they all contributed to a more refined method that is more feasible for real-world testing than the first iteration. With all the design implications processed, a more feasible crowdsourcing method consists of the following phases.

1. **Filtering:** Filtering out the reviews that are unusable due to spam or irrelevancy. Reviews that convey useful information will be marked as helpful.
2. **Fragmentating:** After the automatic sentence splitting of helpful reviews, conducting the same filtering process as the previous phase but this time applied to the resulting review fragments.
3. **Categorizing:** Where the helpful review fragments are categorized in four of the main requirements categories when possible. Fragments that do not cover these four main categories are caught in a fifth category, where they can be manually reviewed or used as input for further phases.
4. **Refining:** A phase indicating that results from the previous phases can be refined even further by designing more specific classification tasks.

The final method with the phases described above is shown visually in Figure 19 on the next page. The method in this exact form will be used for the large scale tests to check the feasibility of a crowdsourcing approach for the elicitation of requirements and for possible further refinement.

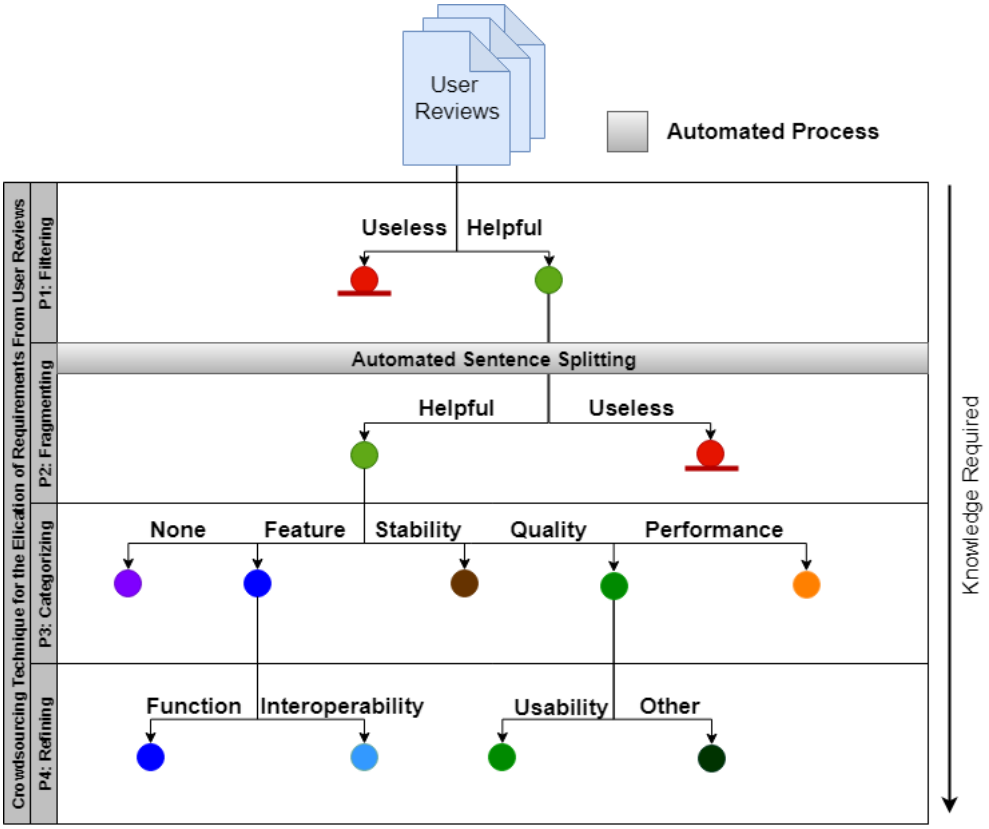


Figure 19: Visualization of the final iteration of the method that will be used for testing.

4 Large Scale Testing

In addition to the creation of a crowdsourcing method for the elicitation of requirements from online reviews, an element of feasibility is also included. As already mentioned in section 1.3, this research goes beyond the creation of a conceptual method and will try to gather evidence that supports the feasibility of crowdsourcing solutions for this purpose. While the internal tests during the design of the method showed promising results, they were conducted in an environment not representative for a crowdsourcing setting and are therefore insufficient to draw generalizable conclusions from. In order to be able to draw generalizable conclusions, larger real-world tests will be conducted to assess the actual performance of the constructed method and to measure the capabilities of the involved crowd workers.

This chapter describes in detail the approach of the large scale tests, the data that will be used and will show the exact configuration of each test. Furthermore, it will discuss the most important aspects that have to be taken into account when the results yielded from the tests will be processed.

4.1 General Approach

Multiple large scale tests will be executed to put the final iteration of the method, as described in section 3.5, to the test. The final iteration consists of three explicit phases and an added fourth conceptual phase. In accordance with the research questions defined in section 1.3, conducting tests with the first three phases only will be sufficient to test the earlier defined hypotheses. Results from the first two phases will indicate whether crowd workers are able to distinguish between useful and useless results. Additionally, the results for the third phase will show the capabilities of the crowd to classify useful reviews into different requirements categories correctly. To clarify, all phases will be tested separately, so a total of three tests will be conducted. To properly test the performance of the entire method instead of the individual phases, the tests will share the same input and will make use of the results outputted by the crowd from the preceding phases.

Earlier we selected the Figure Eight platform as the most suitable candidate for potential real-world testing, which will therefore be the platform used for the three tests. It will make use of its available quality control mechanisms, but will initially not be too restrictive towards participants as long as there is no clear motive for. The aim is to keep the strictness of the quality controls consistent between the three tests, as long as they are sufficient to inhibit crowd workers from cheating or to prevent other causes that will severely hurt the validity of this research from occurring. Initially, the tests will be launched into two separate sessions to reduce the impact from potential unforeseen circumstances. Each test will start with the launch of a session containing only a relatively small part of the input, to explore the course of the test and to see whether it matches expectations. The second session with the remaining part of the input will be launched in an identical fashion, should no indication for major flaws in the method or indication of malicious intent or exploits from the crowd could be established. For the analysis of the results, the outcome will be presented individually for both sessions, but an aggregation of both will also be provided to elucidate the outcome in its entirety. Subsequently, Figure Eight asserts a job limit of a maximum of 1,000 data rows for trial accounts which will likely require the tests to be split between several accounts to work around said limit. The exact configurations for all three tests will be discussed in detail in Section 4.4.

In order to be able to compare the performance of the crowd, a golden standard will be created by having the researcher perform all necessary classifications that the crowd will perform as well. In other words, the researcher will perform the same tasks as the crowd in all three tests. The results of the tests will be analyzed in the context of how the crowd performed compared to the researcher and to see to what degree they match. Identifying the exact areas where the judgments of the crowd do not match with the researcher will allow for the deduction of points where the method could be improved.

4.2 Dataset

A set of real world reviews will have to be involved in the tests to see how the constructed method handles user generated feedback. To get a wide variety of reviews from a multitude of different

software products, the decision was made to reuse a dataset that had been used in other user review related studies. The dataset in question originates from a study conducted by Groen, Kopczyńska, et al. (2017), where the authors instructed five people to manually annotate 360 reviews on software product qualities. The set of products and their respective categories are shown below in Table 12.

Category	Products Groen et al.	Selected
Entertainment	Disney Movies Anywhere	×
	Cleverbot	×
Productivity	Microsoft OneNote	✓
	Tiny Scan Pro	✓
Social Media	TweetCaster	✓
	TweetCaster Pro	✓
Messaging	Viber	✓
	IM+ Pro	✓
Games	PAC-MAN 256	✓
	Sonic & SEGA All-Stars	✓
Smart Products	Philips Hue	×
	August Smart Lock	×

Table 12: The selected products of which the reviews will be used in the upcoming tests.

The choice for the product categories shown in Table 12 are based on the findings of Pagano and Maalej (2013). They identified these categories as the ones that accumulate the most amount of reviews, as they usually contain apps that users interact with frequently and build a relationship with (Pagano & Maalej, 2013). The only exception of this rule however is the Smart products category, which was added by the authors of Groen, Kopczyńska, et al. (2017). As the perceived benefit of that category could not necessarily be extrapolated to this study, the decision was made to remove said category from the dataset. Furthermore, Groen, Kopczyńska, et al. (2017) discovered that the apps from the entertainment category provided disproportionately less valuable reviews compared to the other products. Based on said notion, the decision was made to cut this category altogether as the dataset contained other products for entertainment purposes, as represented by the games category. Lastly, the dataset gathered reviews from the Apple App Store, the Google Play Store and Amazon.com. The reviews contributed by Amazon.com were substantially lower (5,600 out of 132,000), to a degree where the necessity of the inclusion of Amazon.com could be called into question. The purpose of the tests are not focused on researching differences in reviews between multiple sources. Additionally the Amazon.com App store has evolved and grown enormously after the date of the most recent included review (late 2015), rendering the apps likely unrepresentative for the current shape of the Amazon.com platform. Because of these reasons, the decision was made to exclude the reviews from Amazon.com.

To conclude, the dataset that will be used for the tests consists of reviews gathered from eight products from four different categories. The products consist of four free apps and four that require a payment to access, distributed as one of each per category. The modifications to the dataset described above reduced the available user reviews from roughly 132,000 to 114,000 reviews.

4.3 Sample Generation

Due to the limited budget made available to use, it is currently unrealistic to involve all 114,000 reviews from the dataset in the tests. Therefore, a sample will have to be generated that is representative for the characteristics of the entire dataset. The job size limit of a trial account on the Figure Eight platform is currently set at 1,000 data rows, so the sample will match that limit initially. The sample can be expanded at a later stage should there be any reasonable cause for.

The following sections will describe the sample generation process, present and compare the characteristics of both the sample and the entire dataset and will provide arguments as to why the sample is representative.

4.3.1 Products and Reviews

Product	Itunes	Google Play	Total Nr. Reviews	Proportion Total
Microsoft Onenote	6772	2959	9731	9%
Tiny Scanner Pro	2140	601	2741	2%
TweetCaster Pro	724	4023	4747	4%
TweetCaster Free	7758	4050	11808	10%
Viber	67671	4306	71977	63%
IM+ Pro	193	2215	2408	2%
PAC-MAN 256	2261	4402	6663	6%
Sonic & SEGA All-Stars	3548	420	3968	3%
Total:	**91067**	**22976**	**114043**	**100%**

Table 13: Applications and the amount of reviews that are included in the provided dataset.

The entire dataset consists of over 114,000 reviews originating from the eight products are depicted in Table 13. On average, around 80% of the reviews are from the Apple App store and 20% are from the Google Play Store, though this ratio varies a lot between individual products. The Viber communication app is by far the largest contributor of reviews to this dataset, taking up a total of 63% of all reviews.

The size of the sample will depend on the limit imposed by the Figure Eight platform for trial accounts, so a sample of 1,000 reviews will be created. The reduction to that size will be based on the proportion of the reviews that each product contributes to the entire set. However, basing the reduction purely on these proportions will cause problems due to the relative size of the Viber application. Having 63% of your sample consisting of reviews from one single app will leave too little space for the remaining apps, making it unlikely that their selection of reviews will be representative. A correction was therefore administered, limiting the proportion of reviews from Viber to a maximum of 30%. This prevents the contribution from smaller products such as Tiny Scanner Pro from becoming negligible otherwise.

The reviews were selected on the basis of systematic stratified sampling. In this type of sampling, stratified refers to the idea each product is considered a separate subgroup to ensure that all products will be included in the final sample. Systematic means that a selection will be made every *Nth* review from each subgroup (product). In this sample creation process, N is determined by dividing the number of reviews by their maximum allocated space in the sample. This is also applied between the different sources to make sure that for each product, both reviews from the Apple App store and the Google Play store are included. This type of sampling will also ensure that reviews are selected over the entire timespan of the original dataset, minimizing the chance that only reviews will be selected from a product when it was currently not working or unavailable. In other words, it will prevent the inclusion of reviews that were only relevant for a specific and small amount of time compared to the entire lifetime of the product.

Product	Itunes	Google Play	Allocated Space	Proportion Total
Microsoft Onenote	113	49	162	16%
Tiny Scanner Pro	36	10	49	5%
TweetCaster Pro	12	67	79	8%
TweetCaster Free	129	67	196	20%
Viber	*282**	*18**	*300**	*30%**
IM+ Pro	3	37	40	4%
PAC-MAN 256	38	73	111	11%
Sonic & SEGA All-Stars	59	7	66	7%
Total:	**672**	**328**	**1000**	**100%**

Table 14: The number of reviews and their proportion to the total for each application selected for the tests.

Table 14 shows the distribution of reviews for the created sample by using the systematic stratified sampling technique. Highlighted in the table is the correction made for the Viber communication

app, limiting its maximum proportion to 30%. The remaining 700 spaces were distributed between the other seven products, based on their proportion in the entire dataset. The applied selection method successfully ensured that the smaller apps are notably better present in a set of only 1,000 reviews. Furthermore it kept the order from largest proportion to smallest proportion intact, meaning the product with the largest number of reviews is still the largest product in the sample, with the others following suit in the exact same order as the whole dataset.

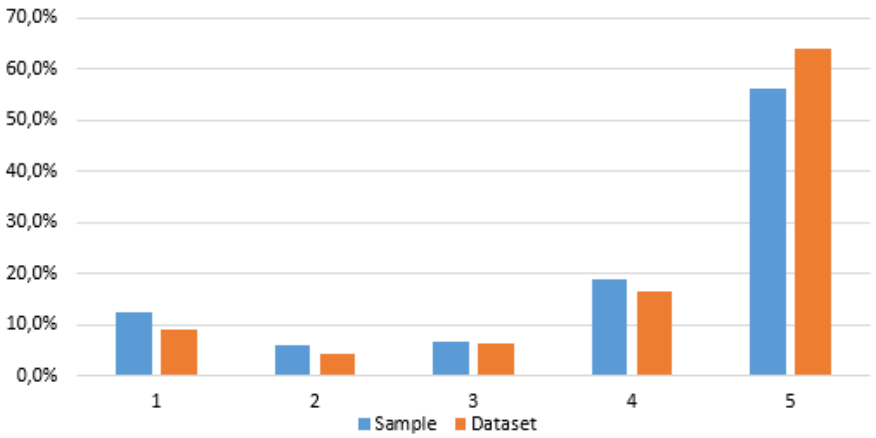
4.3.2 Sentiment Analysis

In this section, the sentiment of the reviews included in the sample will be compared to the sentiment of the entire dataset. Each review from the dataset is accompanied by a star rating portraying the sentiment of the author in a numerical score. It allows us to derive the overall opinion of users of each product and to see whether this overall opinion matches with the average scores in the sample. In Table 15 the average scores provided by the authors of the reviews are depicted for each product.

Average Ratings	**Sample**	**Full Dataset**
Microsoft Onenote	3,81	3,78
Tiny Scanner Pro	4,74	4,72
TweetCaster Pro	3,76	3,92
TweetCaster Free	3,55	3,66
Viber	4,48	4,44
IM+ Pro	3,63	3,71
PAC-MAN 256	4,52	4,54
Sonic & SEGA All-Stars	2,84	2,84
Weighted averages:	**4,04**	**4,19**

Table 15: The average ratings that the products received for both the sample and the entire dataset.

The total average ratings between the sample and the original dataset depict a difference of 0.15 between the sample and the whole dataset. Taking a closer look at the individual products, differences are even smaller or even nonexistent between the two datasets. The largest differences between ratings can be observed at the TweetCaster products with a maximum disparity of 0,16 for TweetCaster Pro. Table 15 shows us that the overall sentiment of the reviews in the sample is very close to the overall sentiment of the entire dataset.



	Overall		Itunes		Google Play	
Stars	**Sample**	**Dataset**	**Sample**	**Dataset**	**Sample**	**Dataset**
1 Star	12,4%	9,0%	12,3%	7,7%	12,5%	14,2%
2 Stars	5,9%	4,2%	3,1%	3,3%	11,6%	7,5%
3 Stars	6,5%	6,4%	5,2%	5,6%	9,2%	9,7%
4 Stars	18,9%	16,5%	18,0%	16,1%	20,8%	18,1%
5 Stars	56,3%	63,9%	61,4%	67,3%	45,9%	50,6%

Table 16: Distribution of star ratings for both the sample and the entire dataset.

Analyzing the distributions of the number of star ratings between the two datasets, a similar story emerges. Table 16 and its accompanying figure depict the percentage of different star ratings for both datasets. As shown, reviews with 5-star ratings are by far the dominant group with a similar trend between the two datasets. No large derivations from these observations can by identified even when distinguishing reviews from the two separate sources. Concluding, the overall ratings and star distributions between the two datasets are close enough to each to deem the sample representative for the entire dataset in regards of sentiment.

4.3.3 Time Analysis

Year	Sample	Dataset
2011	4,9%	2,88%
2012	24,2%	24,72%
2013	18,2%	25,40%
2014	14,7%	18,03%
2015	38,0%	28,97%
Total	**100,0%**	**100,00%**

Table 17: Years of origin of the user reviews for both the sample and the entire dataset.

The final aspect that will determine whether the sample is representative is the analysis of the time of posting of the reviews. Table 17 shows the distribution of the years of origin from the reviews for both datasets. It shows that only a very small amount of reviews were submitted in the year 2011 and that the most recent category (2015) contains the largest amount of reviews. While the sample contains around 9% more reviews from the year 2015 than the entire dataset, these characteristics of different group sizes remain intact.

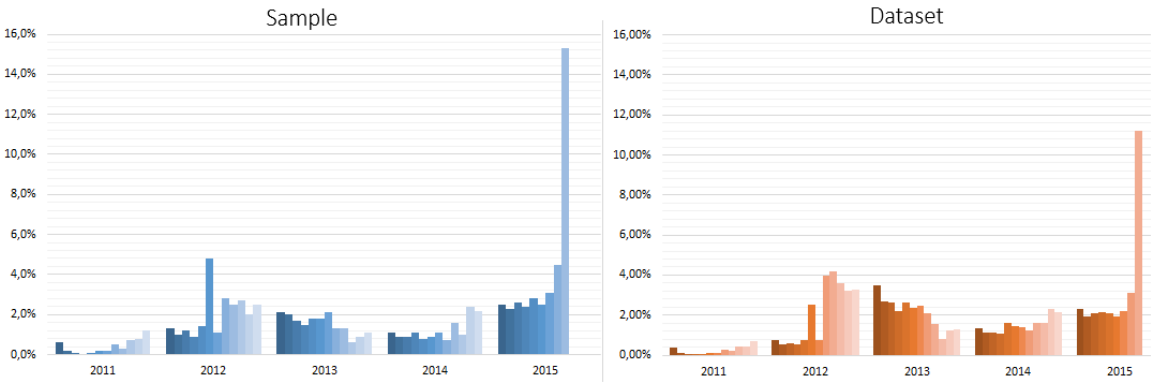


Figure 20: Distribution of the number of reviews over the course of five years.

Lastly, Figure 20 shows the distribution of the reviews for both datasets over the course of 5 years. A similar distribution can be observed as shown in table 17, showing the year 2015 as the main source for the reviews.

Combining the number of reviews per product and their proportion to the total, the overall sentiment of the reviews between the two datasets, and the origin of the reviews over the same timespan,

we have no indication to assume that the sample is not representative for the entire dataset. No major derivations could be identified from the sample compared to the whole dataset, which allows us to conclude that the created sample is of a sufficient quality to produce generalizable results in the to be conducted tests.

4.4 Test Configuration

With the successful creation of a representative sample of reviews available for testing, the detailed configuration of the tests can commence. The logistics of the tests are more impeded due to the limits placed on trial accounts by the Figure Eight platform. To work around these limits, each phase will be tested on an individual account, requiring a total of three accounts for all three tests. Additional accounts will be added should the required workload exceed expectations. Because the tests will be conducted between different accounts, it is important to document all exact settings so that they can be kept consistent.

For quality control mechanisms, participants will be subjected to a eligibility test of 10 questions before being allowed to contribute. The passing rate for the eligibility test is set at 70% which means that participants are allowed to submit three wrong answers. Crowd workers that passed the test are allowed to continue contribution, limited to a maximum of 50 contributions each. This limit is set for the purpose of preventing a small group of early responders to claim all the work. This way, The number of tasks per page is set at ten, meaning that crowd workers will be able to contribute a total of five pages of work to reach the limit of fifty. On each page of work, one quality control question will be randomly appointed inbetween the other reviews. Using these quality control questions, it can be determined whether the answer of a contributor can be tested after they have passed the eligibility test.

The combination of the eligibility test and the quality control questions require a minimum of 15 test questions to be prepared. An even distribution of questions will be prepared for all possible categories of answers, so that crowd workers will be exposed to all possibilities during the eligibility tests. The test questions will be accompanied by small explanations that will be shown for the wrong answers when the test is completed. Using these explanations, crowd workers are able to learn from their mistakes and fine-tune their judgments. These explanations are part of the learning method as described in section 3.1.4.

Furthermore, these tests will not make use of the different contributor levels used by the Figure Eight platform. Figure Eight distinguishes between three levels based on their experience on the platform, with level 1 being the lowest and level 3 being the highest. To not inhibit the chances to create a large crowd of contributors, this function will be excluded in these tests, meaning that all levels of contributors are eligible to at least try the eligibility test. Moreover, contributors that complete a page of work quicker than a limit set beforehand will be automatically disqualified from participating, meaning they tried to cheat or exploit the system by answering randomly. These time limits are set for 20 seconds for phase 1 and 2, and 30 seconds for phase 3. Lastly, all jobs will be provided with the keywords ***App Reviews, Spam Detection and User Reviews*** to advertise the jobs on the platform.

Lastly, in addition to the specific settings for the tests, descriptions were constructed for each individual phase that will provide the knowledge required to complete the work. These descriptions contain an introduction, steps that explain the required workflow and guidelines and examples that will help the crowd workers to make the correct judgments. All phase descriptions are in accordance with the design principles discussed in section 3.1.4 and built further upon the initial descriptions constructed for the internal tests. The following sections will present the content of each phase from that will be provided to the crowd workers.

4.4.1 First Phase

For the first phase, crowd workers are required to classify entire reviews as either helpful or useless. To facilitate this type of work, three parts had to be designed. The first part is the way that the tasks itself are presented to the crowd workers. The tasks were presented to the crowd workers in the following fashion.

Please classify the following review:

"Well done Microsoft!! OneNote is a fantastic program, perfect for day to day stuff and college notes. In fact, everything where a word processor isn't quite the write [sic] tool. I've been waiting for an iPhone app to use OneNote remotely, and this exceeds my expectations!!If you are a OneNote user, I really can't see how you could be disappointed with this perfect companion. Great looking and does the job perfectly!(I had a problem signing in at first, but resubmitted a few times and it worked - now I have no trouble logging in)"

Which category best fits this review? (required)

O Helpful
O Useless

Each task begins with the repetition of the objective of the tasks, represented by the sentence at the top that urges the crowd worker to classify the following review. Following said statement, the review that has to be classified is shown. Afterwards, the crowd worker is asked which category do they think fits the review that they just read, and the available answers are shown. The crowd worker selects its most suitable option and continues to the next tasks. On each page, 10 of these tasks are shown in consecutive fashion.

The crowd worker will decide their answers based on a description that is provided to them. Crowd workers have access to this description throughout the entirety of their job and can be accessed through a collapsible button at the top of each page of work. The provided description for phase 1 is shown on the following page.

Description
In this job, you will be presented with text from user reviews from mobile app stores such as the Google Play Store or the Apple App Store. User reviews can hold important information for developers: feedback from users is important for the evolution and improvement of their application. Before developers gain access to the valuable information that user reviews offer, they require some pre-processing.

The goal of this job is to filter out the spam and to remove useless reviews. This entails classifying individual reviews as either helpful or useless, based on the guidelines below.

Steps

1. Read each user review carefully.
2. Determine whether the review could be of any help to a developer based on the guidelines and examples listed below.
3. Mark the reviews as helpful or useless.

Continue to the next review and repeat steps 1-3.

Guidelines
For each of the presented user reviews, carefully consider whether they can be helpful for developers or not.

- **Useless Reviews**
  - Contain spam or other unrequested or unwanted messages.
  - Their content does not relate to the app and its functions.
  - Express user feelings without elaborating how the app causes those feelings.
  - Don't appear to be written genuinely and contain jokes or absurd statements.
- **Helpful Reviews**
  - Specifically mention aspects of the apps, that is functions, features and behaviour of an app.
  - Report bugs and performance issues that the user encountered.
  - Request potential functions to be added or changes to be made.
  - Provide feedback on recent changes and updates.
  - Present constructive criticism and/or opinions from users included with specific causes or possible solutions.

Examples

- **Useless Reviews**
  - *"WRSET APP OF THE CENTUREY"*
  - *"I Really Like This!"*
  - *"My kids love it. Thanks"*
  - *"I have been using this app for international calling. It saves me a lot money. Great app..."*
  - *"This is by far the worst app ever. I hate it so much."*
- **Helpful Reviews**
  - *"Newest version crashes when opening"*
  - *"Both the texting and calling functionalities are easy to use."*
  - *"Buggy and unreliable. Does not work often. Signs me out regularly. Won't download movies onto my iPad. Disappointing."*
  - *"Ok, but design could be better, too dark"*
  - *"Fun but so data hungry. Why does it need so much data???"*
  - *"I just wish I could turn off the emojis because I don't like them at all."*
  - *"Good app but it is slow and lags sometimes."*

In addition to the clear and concise introduction and the workflow, the description contains plenty

of guidelines and examples that the worker can refer back to when working. The guidelines aim to encapsulate the core principles of each answer category while the examples aim to show a diverse range of actual reviews that belong in each category.

Lastly, a set of test questions were prepared for the eligibility test. The test questions are all real reviews originating from the dataset, that did not make it into the sample. All reviews that were used as test questions for the first phase are shown in Table 18.

TQ	Reviews used as Test Question	Answer
1	Great game	Useless
2	Amazing I don't know what to say	Useless
3	This version is Slow on iphone 5, unusable.	Helpful
4	Fix the bugs, and you've got a 5 star app. I like you can update multiple twitter accounts, FB all at once	Helpful
5	Great program and ability to sync it across all my devices makes it my favorite	Helpful
6	Real nice and convenient. Skype and whatsapp, look out!	Useless
7	The only one I use It is verry good app. I use it daily. Thanks	Useless
8	I updated it... But this update made the game crash when I want to play a race.Please update it again, double check if the update works very well for the app. And please, don't fail SEGA. :|	Helpful
9	The gesture to swipe right to view the menu is interfering with the scrolling gesture making navigation a pain. Otherwise great ap.	Helpful
10	Awesome I love this game it's really cool u should get it	Useless
11	Best game 4lyfe	Useless
12	The best note app, organizing your note with notebook, sections, pages , and subpages A lot of text editing function, including adding link to your own paragraphs. But this app for phone is not as good as that for Mac, less multimedia function than Evernote for phone.	Helpful
13	Great take.. ...on a classic game	Useless
14	I would love this if it gave me notifications like echofon did. Like when somebody faved my tweet, etc.	helpful
15	It's a great app but still room for improvement.	Useless

Table 18: The reviews used as test questions during the first phase.

Identical to the examples given in the description, the reviews used as test question try to cover a wide range of different reviews that the crowd worker may encounter. When they answered one of these test questions incorrectly, they were shown further explanation as to why their choice was the wrong one. All explanations that accompanied these reviews are shown in Table 19.

Nr.	Explanation	Applied to TQ
1	The review is nothing more than an expression of how the reviewer feels about the application.	1, 2, 6, 7, 10, 11, 13
2	The reviewer mentions that the applications crashes or lags on a specific type of phone.	3
3	The reviewer mentions a specific aspect that they like about the application, that they are able to update multiple Twitter and Facebook accounts at once.	4
4	The reviewer mentions a specific aspect that they like about the application, that they like the ability to synchronise between all their devices.	5
5	The reviewer mentions that the applications crashes and lags after performing specific actions and provides feedback on a recent update.	8
6	The reviewer mentions a specific aspects that they dislike about the application, that the navigation options are not working well.	9
7	The reviewer mentions specific aspects that they like about the application, but also express a need for more multimedia functions.	12
8	The reviewer makes a distinctive request about a desired feature that they are missing in the current version of the application.	14
9	The review is nothing more than an expression of how the reviewer feels about the application and no specifics are mentioned about what the reviewer wants to see improved.	15

Table 19: The explanations that accompanied the test questions for phase 1.

4.4.2 Second Phase

For the second phase, the reviews that were marked as helpful are used as input. Only now, all helpful reviews were automatically split per sentence by using an online available text processing tool. The objective of phase 2 is close to identical to the objective of the first phase, warranting the reuse of the core aspects of the parts from the first phase. Only minor adjustments were applied to bring them more in line with the input of this phase, consisting of individual sentences instead of entire reviews.

For the second phase, the layout for the tasks that are presented to the crowd workers looked like the following.

Please classify the following sentence:

"The phone calls and the chatting is what I utilize most and never had an issue until I just updated the app and I can't seem to send any messages."

Which category best fits this sentence? (required)

- O Helpful
- O Useless

As can be observed above, no major deviations from the task layout are present compared to phase 1. It uses the same structure of repeating the objective, presenting the feedback sentence that requires classification and the available options that the crowd worker has access to.

The entire tasks description that goes alongside these tasks is very similar to the description from the first phase and is shown on the next page. The most important change is the removal of the mention of reviews, which are changed to feedback sentences to make them more in line with the objective of the tasks. Furthermore, the reviews provided as examples themselves are split per sentence as well. The guidelines are kept identical to the guidelines from the first phase. The entire description provided alongside the tasks for phase two can be viewed on the next page.

Description
In this job, you will be presented with sentences derived from user reviews from mobile app stores such as the Google Play Store or the Apple App Store. User reviews can hold important information for developers: feedback from users is important for the evolution and improvement of their application. Before developers gain access to the valuable information that user reviews offer, they require further processing.

The goal of this job is to **filter out the spam and to remove useless sentences**. This entails classifying individual sentences as either **helpful** or **useless**, based on the guidelines below.

Steps

1. Read each sentence carefully.
2. Determine whether the review could be of any help to a developer based on the guidelines and examples listed below.
3. Mark the reviews as **helpful** or **useless**.

Continue to the next review and repeat steps 1-3.

Guidelines
For each of the presented feedback sentence, carefully consider whether they can be helpful for developers or not.

- **Useless Sentences**
  - Contain spam or other unrequested or unwanted messages.
  - Have content not related to the app and its functions.
  - Express user feelings without elaborating how the app causes those feelings.
  - Don't appear to be written genuinely and contain jokes or absurd statements.
- **Helpful Sentences**
  - Specifically mention aspects of the app about its functions, features and behaviour.
  - Mention bugs and performance issues that the user encountered.
  - Request potential functions to be added or changes to be made.
  - Provide feedback on recent changes and updates.
  - Present constructive criticism and/or opinions from users included with specific causes or possible solutions.

Examples

- **Useless Sentences**
  - *"WRSET APP OF THE CENTUREY"*
  - *"I Really Like This!"*
  - *"My kids love it. Thanks"*
  - *"I have been using this app for international calling, it saves me a lot money."*
  - *"This is by far the worst app ever I hate it so much."*
- **Helpful Sentences**
  - *"Newest version crashes when opening"*
  - *"Both the texting and calling functionalities are easy to use."*
  - *"Buggy and unreliable.*
  - *Does not work often.*
  - *Signs me out regularly.*
  - *Won't download movies onto my iPad."*
  - *"Ok, but design could be better, too dark"*
  - *"Fun but so data hungry."*
  - *"I just wish I could turn off the emojis because I don't like them at all."*
  - *"Good app but it is slow and lags sometimes."*

While the previous two parts reuse the majority of the content from the first phase, a new set of test questions had to be created to introduce the crowd workers to feedback sentences. In addition to functioning as a quality control mechanism, they also aim to learn the crowd workers how they can distinguish between helpful and useless feedback sentences. The reviews chosen for the creation of the test questions were selected on that exact premise and can be viewed in Table 20.

TQ	Reviews used as Test Question	Answer
1	Right now nothing in the settings that you can use to assign different / custom text tones.	Helpful
2	There's no sound when someone calls you.	Helpful
3	I deserve it.	Useless
4	Great program and ability to sync it across all my devices makes it my favorite.	Helpful
5	I use OneNote on my laptop and love it.	Useless
6	Please fix this!!	Useless
7	I use DID Groundwire app, Skype, Google Voice and few other apps and services when travel in Europe or Asia.	Useless
8	Only complaint would be the app runs a bit sluggish, for example during launch or switching between tabs and conversations.	Helpful
9	Too Much screen at top now being used!!	Helpful
10	Until then this program is simply a note viewer.	Useless
11	Very disappointed.	Useless
12	Easy to use, no watermark, emails fast.	Helpful
13	Amazing.	Useless
14	Simple and easy to use interface.	Helpful
15	I have THREE BARS and full Wi-Fi yet it still won't sync.	Helpful

Table 20: The reviews used as test questions during the second phase.

The explanations that were shown when one of the test questions were answered incorrectly are shown Table 21. In the instance of a wrong answer, the explanations aimed to highlight the part of the sentence that contained the helpful part. This would aid the crowd workers in understanding the error that they made and train them to better recognize the helpful part of a sentence in the future.

Nr.	Explanation	Applied to TQ
1	The review expresses a desire for a new functionality that the user wants.	1
2	The sentence mentions that the user encountered an unexpected problem with a specific function.	2, 15
3	The sentence provides not information that is relevant to an application of any kind.	3
4	The sentence expresses how a user feels about specific functionalities of the application.	4
5	The sentence is nothing more than a generic description of how the user experiences the application.	5, 10, 13
6	While the sentence indicates that a certain problem with the app should be fixed, it does not specify what exactly the problem entails and is therefore not helpful for the developers.	6
7	The sentence is nothing more than a generic description of how the user used the application themselves.	7
8	The sentence mentions that the app is a bit slow and provides examples for situations when this observation occurs.	8
9	The sentence provides a detailed description about what the user feels that is wrong with the application.	9
10	The sentence is nothing more than a generic description of how the user feels about the application.	11
11	The sentence expresses specifically what the user liked about the application.	12, 14

Table 21: The explanations that accompanied the test questions for phase 2.

4.4.3 Third Phase

The parts providing the content to the crowd workers for the third phase required an entirely different approach that only kept the main structure intact. The difference in objective requires crowd workers to distinguish between five different answers instead of two. Providing this amount of options to the crowd workers is more demanding on their knowledge, so the content provided to them should provide this to them as effectively as possible.

Please classify the following sentence:

"It's constantly telling me I'm not authorized! tells me I've reached my 350 max tweets per minute after my FIRST tweet of the day!"

Which category best fits this sentence? (required)

- O Feature Request
- O Stability Feedback
- O Performance Feedback
- O Quality Feedback
- O None of the above

However, to not unnecessarily interfere in the workflow of the crowd workers, the part where the tasks are actually performed are kept as streamlined as possible. The objective of the task is repeated in the beginning, but kept short and concise. The review sentence that requires classification is familiarly placed in the middle and is followed by the clickable possible answer categories.

Much more emphasis however is laid upon the task description, as it tries to compensate for the more demanding nature of the task. In absolute terms, more content was required to properly cover the core principles of the five options. The constructed task description is therefore a bit longer than its predecessors. However, even with extending the length of the task description, only two or three sentences were allocate to explain what each category entailed. A part of the important aspects was therefore shifted to the examples, that try to fill in possible knowledge gaps by providing examples that perfectly represent their category. The entire description that will be utilized for the third test is shown on the following page.

Description
In this job, you will be presented with fragments of user reviews from mobile app stores such as the Google Play Store or the Apple App Store. User feedback is important for the evolution and improvement of apps and can come in many different forms.

The goal of this job is **to identify what type of feedback the review mentions**. This entails classifying individual reviews as either being a **feature request, stability, performance, quality** related or **none** at all based on the guidelines below. If the feedback sentence fits **multiple categories** according to the guidelines, please choose the **predominant** one.

Steps

1. Read each feedback sentence carefully.
2. Determine whether the feedback sentence matches with one of the categories based on the guidelines and examples listed below.
3. Mark the feedback sentence as **Feature Request, Stability Feedback, Performance Feedback, Quality Feedback** or **None** of the above

Continue to the next review and repeat steps 1-3.

Guidelines
For each of the feedback sentences, carefully determine the relevant feedback category.

- **Feature Request:**
  - Expresses a desire for functions to be implemented in the future.
  - Mentions what features or functions are missing according to the user.
- **Stability Feedback:**
  - Refers to crashes or other instances when the application froze or did not work at all.
  - Mentions bugs that completely prevented users from interacting with the application.
- **Performance Feedback:**
  - Mentions how quick or slow the application performs after the user performs specific actions.
  - Refers to the usage of battery, memory or internet data by the application.
  - Note: poor performance leading to a crash is a stability issue.
- **Quality Feedback:**
  - Explicitly mentions what the users liked or disliked about how well/poorly the application does something (not a generic "I love this app").
  - This could include how easy to use the application is perceived by the user.
- **None of the above:**
  - Do not appear to mention information relevant to an application of any kind.
  - Mention other aspects about an application that clearly does not fit with the other previous categories.

Continues on next page.

Examples

- **Feature Request**
  - *"I just wish I could turn off the emojis because I don't like them at all."*
  - *" Add video calling and you might have a contender for Skype."*
  - *"It's nice but lacks the download feature that I have on the app for my android phone"*
- **Stability Feedback**
  - *"Newest version crashes when opening"*
  - *"Buggy and unreliable, does not work often and signs me out regularly."*
  - *"Still not working, keeps crashing after login details are entered."*
- **Performance Feedback**
  - *"Good app but it is slow and lags sometimes."*
  - *"The most recent update improved the speed."*
  - *"Stalls every time you write a tweet."*
- **Quality Feedback**
  - *"Both the texting and calling functionalities are easy to use."*
  - *"It is smart, quick, and compact.*
  - *"Very customizable and clean lay out."*
  - *"You can be friends with who you want and it protects your privacy, I love it!"*
- **None of the above**
  - *"My kids love it. Thanks"*
  - *"Not even worth the download time."*
  - *"This is by far the worst app ever. I hate it so much."*

Similarly to the previous two phases, fifteen test questions were composed that aimed to check whether the crowd workers understood the content of the task description. Each possible answer category was allocated three test questions, creating an even distribution between the different answer categories. The composed test questions and their corresponding answers are shown below in Table 22.

TQ	Reviews used as Test Question	Answer
1	Must have app for smartphones.	None
2	Can't post, refresh anything without error msgs.	Stability
3	Will give it another chance when on screen directional Controls make an appearance.	Feature
4	I'm always on the go and to be able to work on the go is a necessity.	None
5	I open it up and it closes on me.	Stability
6	Pretty useful, just have some security concerns.	Quality
7	Secondly I want to be able to choose text writing on my photos (different colors, fonts and sizes).	Feature
8	I recommend you download and install ASAP.	None
9	This app runs so slowly that it is essentially worthless.	Performance
10	This version is Slow on iphone 5, unusable.	Performance
11	No humming and background noises compared with Skype.	Quality
12	Then it will be a worthwile app, until then, its pretty useless First time I tried to start onenote, 15 seconds passed, and it still didn't finish loading.	Performance
13	What I don't like: Crashes constantly!	Stability
14	Notifications are at the top of my wishlist, though.	Feature
15	Easy user interface and great functionality.	Quality

Table 22: The reviews used as test questions during the third phase.

Explanations were also provided to clear up any possible misconceptions that crowd workers could have derived from the task description up until this point. These explanations and their link to their respective test question can be viewed in Table 23.

Nr.	Explanation	Applied to TQ
1	The user does not mention any specific information that could be helpful for the developers.	1, 8
2	The user describes a problem that completely prevents them from using the app.	2, 5, 13
3	The user makes a specific statement that expresses a desire for a new functionality.	3, 7, 14
4	The user does not mention any information that is specifically related to an app.	4
5	The user describes explicitly what they dislike about the application.	6
6	The user describes that the app is working, but that it is running too slowly to properly use.	9, 10, 12
7	The user mentions explicitly what they like about the application.	11
8	The user describes explicitly what they like about the application.	15

Table 23: The explanations that accompanied the test questions for phase 3.

4.5 Interpreting the Results

Using the exact setup described in the previous sections, the tests will be launched in their intended order. With the exception of phase 1, all phases require the output from the previous phase to continue. To keep track of the quality of the output from each individual phase, separate analyses will be conducted for each phase.

We previously mentioned that a golden standard will be created by a single researcher to compare the outcome of the crowd with. This golden standard will be created by requiring the researcher to complete the entirety of the jobs that the crowd will be performing. For each phase, the analyses will show to what degree the judgments of the crowd match with the judgments of the researcher.

As this golden standard will depend on the judgment of one single researcher, an element of review will be added to the analyses of the outcome of all individual phases. After the first comparison of the judgments of both parties, the answers that did not match will be reviewed. This way, compensation can be applied for any errors in judgment that were made by the researcher. Furthermore, it is possible that certain judgments could only be made when more knowledge was available than provided in the task descriptions. To make it more fair for the crowd, each answer will be reviewed on the plausibility that a judgment could be made correctly on the basis of the provided information. In other words, it will compensate for errors that the crowd made due to the lack of provided knowledge.

In each analysis, both the results from before and after the review will be shown to enable the comparison between them and to provide transparency in the effects of the review process. Both type of results and their implications for the effectiveness of the method will be interpreted for the two different levels of strictness.

5 Results

This chapter will analyze and discuss the generated results from the execution of the tests that are described in chapter 4. This chapter is split into three sections, each discussing one of the three tests respectively. Each section will be separated in three different parts, each discussing a different aspect of the generated results. These aspects are shown below.

1. **Contributor Demographics:** Will provide background information about the contributors who participated in the job. Describing in what way the contributors participated in the job will set the stage to identify and explain potential differences in the outcome of the test.
2. **Job Statistics:** Will explain how the job itself was perceived by the contributors and will explain the trajectory that the job had from launch until completion.
3. **Outcome:** Will dive deeper in the results of the test, the quality of the results and their implications for the method. The performance of the crowd will be compared with the performance of the researcher to see in what way they agree and disagree, to form an indication of the capabilities of the crowd.

Each section includes a brief conclusion, where the most important performance implications are summarized and highlighted, as well as explained and substantiated with either the results, or other qualitative remarks regarding the course of the test.

5.1 First phase

As explained in more detail in chapter 4.3, the first test entails the launch of a job on Figure Eight in two different sessions. The job utilized the multiple customization and quality control settings offered by the platform to manage the way that the crowd would interact with the tasks. The summary of how both sessions were set up by using these options is shown in Table 24.

Session	Nr. of Reviews	Required Judgments	Judgment Limit	Nr. Test Questions	Required Passing %	Cost Per Judgment
First	200	600	50	13	70	$0.04
Second	800	2400	50	15	70	$0.03

Table 24: Summary of the launched jobs for the first phase.

The entire job entailed the classification of 1,000 user reviews, which were divided between a first session of 200 user reviews and a second session of 800 user reviews. Each review required three judgments by three unique contributors, resulting in a minimum of 3000 judgments/contributions required to finalize the job for this phase. In order to prevent that hypothetically only three individual contributors could complete the entire job, a limit of maximum 50 judgments per contributor was established.

Subsequently, 13 test questions for the first session and 15 test questions for the second session were set-up as described in chapter 4.3, to enable the quiz that contributors had to pass before being allowed continue their work. Contributors had to classify 70% of the 10 test questions correctly in order to pass and they were shown the correct answer and a brief explanation when they answered incorrectly. The first session adhered to the by the platform minimum recommended amount of 13 test questions. For the second session two additional test questions were added to offer unique quality control questions on each page of work (10 judgments, maximum of 5 pages per contributor) for contributors that wanted to contribute the maximum allowed amount of 50 judgments.

Lastly, the provided payment for each judgment was set at 4 cents per judgment for the first session and later adjusted to 3 cents per judgment to better fit to the recommended guideline of around $8.00 earnings per hour per contributor. This decision was made based on the notion that the first session took a lot less time to complete than expected. We therefore wanted to explore the potential effects that would occur when the rewards were lowered slightly.

5.1.1 Contributor Demographics

With the settings described above, the job was launched and in turn gathered a large group of respondents, as illustrated in Table 25.

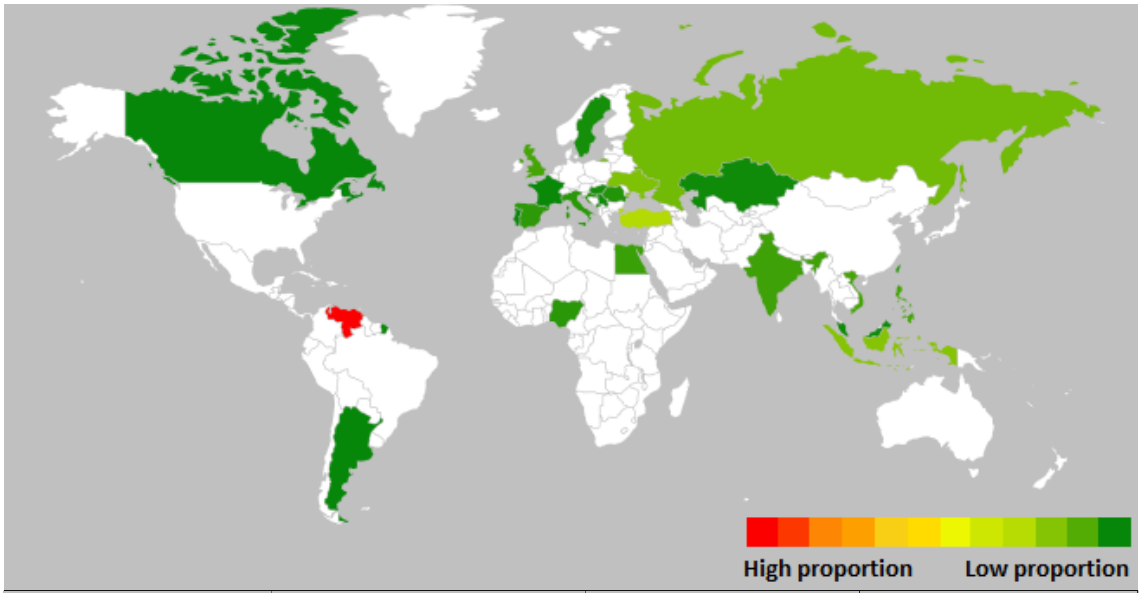
Session	Nr. of Contributors	Test Passed	Test Failed	Average Trust Level	Average Contributions	Average Testquestions	Avg. Incorrect Testquestions
First	50	42	8	79,37%	20,3	9,8	1,9
Second	127	118	9	89.18%	31,1	12,1	1,2

Table 25: Involved contributors and their performance for both job sessions.

The first session gathered a crowd of 50 contributors, of which 42 passed the initial test questions and were allowed to continue their participation. Eight people were filtered out, as they couldn't demonstrate that they possessed the required knowledge to participate properly in this job. The 50 contributors had an average trust level of around 79%, which is the value that indicates the reputation of the contributor based on the performance of other completed jobs in the past on the same platform. The contributors of the first session contributed an average of 20,3 judgments before quitting the job and made on average 9,8 test questions. Out of all test questions, 1,9 of them were answered incorrectly on average.

Furthermore, the second session gathered a crowd of 127 contributors. Of this crowd, 118 of them passed the initial test questions and 9 of them failed and were excluded from further participation. The crowd workers contributing to the second session had an average trust level of around 89% and contributed an average of 31,1 judgments. Regarding the test questions, on average 12,1 were made per contributor and only 1,2 were answered incorrectly.

Compared to the first session, the relatively lower amount of contributors who didn't pass the test could be explained by the overall higher level of trust, which indicates that these contributors have more experience in crowd work or have a better general level of knowledge better suited to this kind of job. The potential higher level of knowledge can be substantiated with the lower amount of incorrectly answered test questions. Finally, more test questions on average were answered in the second session due to the higher amount of average contributions, meaning that contributors from the second session progressed further into the job than contributors from the first session.



Country	Contributions	Country	Contributions	Country	Contributions	Country	Contributions
Venezuela	1407 (28,7%)	United Kingdom	190 (3,9%)	Taiwan	90 (1,8%)	Serbia	40 (0,8%)
Turkey	512 (10,4%)	India	170 (3,5%)	Romania	90 (1,8%)	Malaysia	33 (0,7%)
Indonesia	379 (7,7%)	Egypt	160 (3,3%)	Kenia	50 (1,0%)	Kazachtstan	32 (0,7%)
Ukraine	359 (7,3%)	Spain	120 (2,4%)	Hungary	50 (1,0%)	France	20 (0,4%)
Russia	320 (6,5%)	Italy	110 (2,2%)	Marocco	50 (1,0%)	Portugal	13 (0,3%)
Philippines	190 (3,9%)	Nigeria	109 (2,2%)	Sweden	50 (1,0%)	Argentina	10 (0,2%)
Vietnam	190 (3,9%)	Croatia	100 (2,0%)	Montenegro	50 (1,0%)	Canada	10 (0,2%)

Table 26: Visualization of the contributions per country for phase 1.

The countries of origin from all involved contributors in this job tell an interesting story as well, as shown in the visualization that accompanies Table 26. While the job received contributions from places all over the world, the vast majority of contributions were made from people in Venezuela. The number of contributions received from Venezuela is nearly three times as large as the second largest group of contributors, originating from Egypt. When grouped together, countries from eastern Europe and/or Russia also appear to be quite interested into crowd work. Distinctions between time zones or other differences from these results can't really be made or observed. We will therefore focus on the contributors themselves or the channels that they came from from this point onwards.

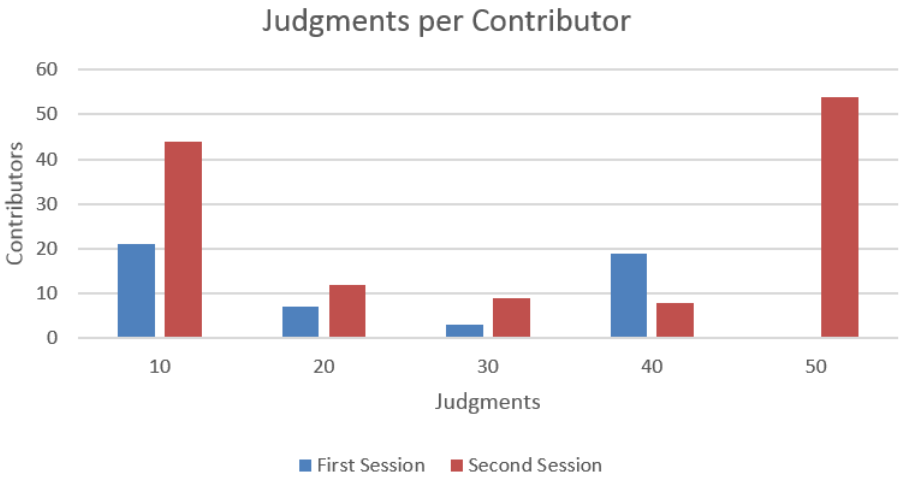


Figure 21: The differences between the amount of contributions made per contributor.

Figure 21 illustrates a divide between the crowd workers, as contributors appear to gravitate towards either doing the minimum or the maximum possible amount of contributions. For both the first and the second session, large groups of contributors stopped working on the job after

submitting only 10 judgments, as a clear drop off can be observed after the first 10 judgments.

Although nobody from the first session contributed more than 40 judgments, the largest amount of contributions appear to come from contributors who reach the maximum allowed amount of 50 judgments. It is possible that the small amount of required judgments from the first session, in combination with the relatively short time that the session was active prevented contributors from going beyond the 40 judgments mark, as the available judgments could have ran out at that point. Nevertheless, it appears that this kind of work in this form has the capability to keep workers interested for an extended period of time.

	First Session (200 Reviews)		**Second Session (800 Reviews)**	
Channels	**Contributors**	**Avg. Trust level**	**Contributors**	**Avg. Trust level**
ClixSense	25	80%	49	93%
FigureEight	19	84%	40	85%
NeoBux	9	87%	28	87%
InstaGC	3	74%	3	86%
GetPaid	2	81%	3	97%
SuperRewards	-	-	1	100%
TimeBucks	-	-	1	92%
Wannads	-	-	1	92%
Entropia	-	-	1	90%
Overall	**58**	**82%**	**127**	**89%**

Table 27: Source channels of the contributions and the average trust levels of their contributors.

The Figure Eight platform distributes launched jobs to different channels that they are partnered with, in order to reach a large amount of available crowd workers that potentially want to contribute. Table 27 shows from which channels the contributors originated from in this job for both sessions. It also depicts the average trust levels of those contributors per channel to give an indication of the overall capabilities of these contributors.

The majority of the contributions came from ClixSense, NeoBux or the Contributor Portal of Figure Eight itself, which appear to be the three largest and most active channels. The second and larger session of the two had unique contributions received from channels such as Entropia, SuperRewards, TimeBucks or Wannads. They appear to be the smaller or slower moving channels, as none of these were reached, or acquired a response from a crowd worker in the time that the first session took to reach completion. Lastly, a small difference can be observed between the average trust levels between the channels. According to this data, the smaller channels have very large fluctuations and are not consistent in the quality of crowd worker that they offer. This effect is less prevalent for the bigger channels, however similar differences can be observed between the two sessions from the ClixSense channels. Nevertheless, the overall quality of all involved contributors from all different channels appear to be sufficiently trusted, which in turn means that the workers who contributed to this job have some form of experience with this kind of work.

5.1.2 Job Statistics

The Figure Eight platform automatically deploys a survey to the contributors after they have finished their tasks. This survey enables us to gain insight into their perception of the job and how they experienced the instructions, the test questions, the ease of the job and the overall payment provided for the work. The contributors were not required to provide this feedback as the survey was optional. The results from this automatic contributor survey are presented in Table 28.

Session	**Respondents**	**Instruction Clarity**	**Test Question Fairness**	**Ease of Job**	**Pay**	**Overall**
First	10	4.0 / 5	4.0 / 5	3.9 / 5	3.4 / 5	**4.2 / 5**
Second	17	4.4 / 5	4.2 / 5	4.1 / 5	4.2 / 5	**4.3 / 5**

Table 28: Contributor perception of the tasks.

With only 20% of contributors from the first session and only 13% of contributors of the second

session participating in this survey, general response can be seen as relatively low. Nevertheless, the different feedback categories and their average scores still tell an interesting story.

With an overall rating of 4.2 and 4.3 for the first and second sessions respectively, we can conclude that the contributors were left with an overall positive feeling about the job after completion. Additionally, the instruction about the job (as presented in section 4.3.1) were perceived as very clear and appear to fit the job activities very well. The test questions were perceived as very fair, which is supported by the negligible (<1%) amount of contentions made on the test questions. Subsequently, with a average score of 4 out of 5 regarding the ease of the job for both sessions, it can be concluded that the contributors did not think that the job was too hard. Finally, the score for payments conveys an interesting message that was contradictory to our expectations. The contributors were less satisfied with the pay for the first session, even though the pay was higher than the payment in the second session (4 cents per judgment versus 3 cents per judgment respectively). Nevertheless, the ratings indicate that the contributors perceived payment to be fair for this kind of job.

With this overview of the contributor perception of the job, we will continue with the analysis regarding the size, the completion time of the job and its total cost.

Session	Date	Launch	Finished	Duration	Review Judgments	Untrusted Judgments	Test Question Judgments	Total Judgments	Total Cost
First	07/05/2019	10:47	11:16	29 min.	600	9	477	1086	$33.60
Second	15/05/2019	11:27	12:49	82 min.	2400	18	1437	3855	$97.20

Table 29: Launch time and completion statistics for the jobs in this phase.

Both sessions were launched a bit more than a week from each other at a similar time of the day. The first session took 29 minutes to complete and the second session took 82 minutes, which means that the larger session took less time to reach completion in proportion to the first session. To reach completion, the sessions required 600 or 2400 valid judgments after the contributors passed the initial quiz and its test questions. On top of the valid judgments, an additional 477 and 1437 judgments were made to complete the quiz and the quality control question for both sessions. These quality control questions caught 27 judgments made by contributors who passed the initial quiz, but failed the control questions distributed throughout the rest of the task. The system therefore labelled them as untrusted and even though they received payment for those, are not included in the final results. The required judgments for the reviews, the test and quality control questions and the untrusted judgments bring the total judgments for completion to 1086 and 3855 for the individual sessions, or a grand total of 4941 judgments for the entire job. This total amount of judgments came down to a total cost of $130,80 for 1,000 user reviews, with the 20% usage fee of the platform included.

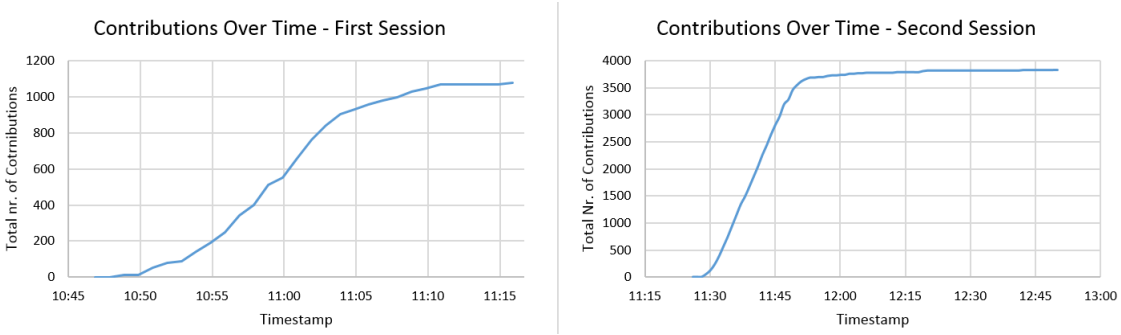


Figure 22: Distribution of received contributions over time.

As shown in Figure 22, the cumulative distribution of contributions received over time show an interesting view of the course of the jobs after their launch. From their time of launch, the jobs required a small amount of time to gain traction, presumably because it take a couple minutes for contributors to receive, understand the objective and set up the job. After the first minutes, the contributions quickly started accumulating to the point where the majority of the job was completed roughly around the halfway mark. This can clearly be observed from the graph of the

second session, as it depicts a quick rise in contributions received in the first half an hour. This is followed by a long tail where only a few contributions are received during the second half of the job duration. It can therefore be concluded that the total duration of each session is somewhat misleading, as the majority of the work is already completed in the first half.

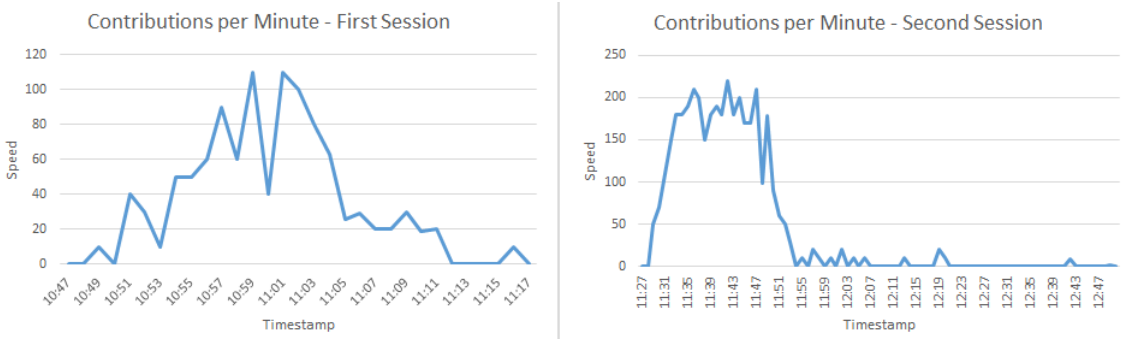


Figure 23: Contributions received per minute for both sessions.

This is further substantiated by looking at the speed of the contributions received per minute during the job duration, as depicted in Figure 23. The speed of contributions received quickly ramped up in the first couple of minutes following the job launch, before becoming somewhat stable during the halfway point, followed by a quick drop-off towards the finalization of the job in the final minutes. However, a difference can be observed between the first and second session, as the first session is more evenly distributed than the second. This can likely be explained by the smaller size and the shorter duration of the job in the first session. It therefore reaches the finalization phase already much sooner before it can reach its full potential and therefore has its speed peaking much lower than the second session.

Session	Average Speed	σ Speed	Average Time per Contributor	σ Average Time
First	38 Judg. per Min.	29,8	13,7 Seconds per Judgment	15,3
Second	46 Judg. per Min.	74,1	12,6 Seconds per Judgment	15,1

Table 30: Average speed of the job and average time per judgment per contributor

Finally, Table 30 shows the average speed of contributions received per minute during the entire course of the job and how much time it took on average for a contributor to reach and submit a judgment. On average 38 judgments were received per minute during the first session and 46 judgments per minute for the second session, further indicating that the second session reached a much higher speed despite the longer required time towards completion. The relatively high standard deviation however indicates that the speed fluctuated quite a lot during the duration of the job.

For the contributors, it took them on average around 13 seconds to reach a judgment for a single review. For quality reasons, the job was limiting contributors to 20 seconds per page of work (10 judgments) to filter out contributors who judged randomly. The actual average times indicate that contributors took much longer to complete a page of work than the expected 20 seconds. The standard deviation however shows again that there is quite a big difference between the capabilities of individual contributors, as there are contributors who both took much more and much less time on average to complete a page of work.

5.1.3 Outcome

This section explores the differences and similarities of the classifications by the crowd and the researcher by comparing them. It therefore presents the quality of the results, both before and after the review process by the researcher, which is described in detail in section 4.5.

	Researcher		Crowd		Researcher		Crowd	
Session	**Helpful #**	**Useless #**	**Helpful #**	**Useless #**	**Helpful %**	**Useless %**	**Helpful %**	**Useless %**
First	78	122	102	98	39%	61%	51%	49%
Second	302	498	406	394	37,75%	62,25%	50,75%	49,25%
Overall	380	620	508	492	38%	62%	50,8%	49,2%

Table 31: Comparison of helpful and useless reviews identified by the researcher and the crowd.

Before the review process has taken place, the researcher classified 38% of the reviews as helpful and 62% as useless out of the sample dataset of 1,000 reviews. In comparison, the crowd identified nearly 51% of the same reviews as helpful and close to 49% as useless. Although these numbers are consistent between the two different sessions, they indicate a relevant discrepancy between the two classifications and require a more thorough analysis.

Session	**True Positives**	**Actual TP**	**True Negatives**	**Actual TN**	**False Positives**	**False Negatives**	**Precision**	**Recall**
First	93	122	73	78	5	29	0,95	0,76
Second	366	498	274	302	28	132	0,93	0,73
Total	459	620	347	380	33	161	0,93	0,74

Table 32: Detailed comparison of the results of the crowd before the review.

Table 32 shows to what degree the answers of the crowd correctly matched with the judgment of the researcher. In Table 32, the Actual True Positives and the Actual True Negatives represent the classification of the researcher, whereas the other values refer to the classification of the crowd. In this context, Positives mean the reviews that were classified as useless, as the objective of this phase of the method is to filter out as many useless results as possible. Subsequently, the Negatives refer to the reviews that classified as helpful and will stay in the dataset for further usage in the following phases.

The True Positives (TP) and the True Negatives (TN) are the amount of reviews that match the judgment of the researcher, and are therefore seen as being classified correctly by the crowd. The difference between the TP of the crowd and the actual TP of the researcher however is substantial, as the crowd classified 459 out of the 620 reviews of the researcher as useless. This is further substantiated by the False Negatives value, which indicates that the crowd in total marked 161 reviews as helpful when they were marked as useless by the researcher. This effect is summarized in the recall value of 0,74, which means that the crowd was only able to find 74% of the useless reviews that the researcher found, leaving on average 26% of all useless reviews in the dataset.

For the reviews that were marked as helpful, the results tell an entirely different story. The True Negative values show that the crowd was able to identify 347 out of the 380 helpful reviews that the researcher identified, with only 33 reviews being incorrectly marked as useless. This results in the precision value of 0,93, which means that the first phase of this method only incorrectly discards 7% of helpful reviews and that 93% of helpful reviews remain in the dataset.

After this point, the second reviewing process was initiated by the researcher, meaning that the reviews that had no alignment between the judgments of the researcher and the crowd were reconsidered and judged for a second time. This process is described in detail in section 4.5, but from the perspective of the researcher means that previous judgments had to be reconsidered and checked for classification errors on the part of the researcher. The effects of this reviewing process and the implications for the previously discussed results will be discussed below.

	Researcher		Crowd		Researcher		Crowd	
Session	**Helpful #**	**Useless #**	**Helpful #**	**Useless #**	**Helpful %**	**Useless %**	**Helpful %**	**Useless %**
First	91	109	102	98	45,5%	54,5%	51%	49%
Second	362	438	406	394	45,25%	54,75%	50,75%	49,25%
Overall	453	547	508	492	45,3%	54,7%	50,8%	49,2%

Table 33: Comparison of helpful and useless reviews by both parties after the reviewing process.

After the reviewing process, the researcher had marked 453 out of 1,000 reviews as helpful, indicating a shift of 73 reviews from useless to helpful. For these 73 reviews, it meant that the researcher

either had to concede that the value of the review was initially misjudged or that it was unfair to expect from the crowd to recognize the uselessness of certain reviews, based on the provided descriptions and guidelines. This bring the proportion between helpful and useless reviews for the two involved parties much closer to eachother. The effects that the changed proportion between helpful and useless reviews had on the effectiveness of this phase is described below in Table 34.

Session	True Positives	Actual TP	True Negatives	Actual TN	False Positives	False Negatives	Precision	Recall
First	94	109	87	91	4	15	0,96	0,86
Second	366	438	334	362	28	72	0,93	0,83
Total	**460 (+1)**	**547 (-73)**	**421 (+0)**	**453 (+0)**	**32 (+0)**	**87 (+0)**	**0,93 (+0)**	**0,84 (+0.10)**

Table 34: Detailed comparison of the results of the crowd after the review.

The effect of the reviewing process is most notable for the recall values, as it increased from 74% to 84% overall. This is explained by the decrease in False Negatives after the review, as the researcher had to concede that 74 reviews were unfairly classified as useless during his first initial judgment. This increase in recall value shows that this first phase of this method is 84% effective in removing useless reviews from the dataset instead of 74%, depending on the leniency of the judgment of the researcher.

Additionally, the very minor change in False Positives (one single review unfairly judged as helpful), means that the reviewing process had no visible effect on the precision of the method. Therefore no notable changes can be observed regarding the capabilities of the method to recognize helpful reviews.

Finally, the results will be compared between the reviews where contributors reached a full agreement (unanimous judgment) separately from each other, or where they were indifferent and only two out of three contributors agreed. The results from the comparison between these two different categories can be observed in Table 35.

Session	Reviews	Unanimous Judgments	Correct Judgments	Incorrect Judgments	Indifferent Judgments	Correct Judgments	Incorrect Judgments
First	200	159	153	6	41	28	13
Second	800	585	532	53	215	147	68
Total (%)	**1000 (100%)**	**744 (74,4%)**	**685 (92,1%)**	**59 (7,9%)**	**256 (25,6%)**	**174 (67,9%)**	**82 (32,1%)**

Table 35: Comparison between full agreement and indifferent judgments.

As shown in Table 35, the contributors were able to reach a full agreement for 744 out of the total 1,000 reviews from the dataset. For 256 of them, only partial agreement was observed so the judgment was based on the majority vote between the three contributors. Out of the full agreement category, 92,1% of those reviews turned out to be classified correctly compared to the judgment of the researcher after the reviewing process. In contrast, from the category of reviews where only partial agreement was reached, roughly 68% had received judgments that were in alignment with the judgments of the researcher.

These results indicate that reviews that received the same classification from three different contributors are most likely to have received the correct judgment. Nevertheless, while the amount of incorrectly judged reviews is much higher where the contributors were indifferent, more than two thirds of those 256 reviews still received the correct judgment. Indifferently judged reviews therefore do not automatically mean that they are incorrect, but it can be reasonable to be more cautious when drawing conclusions from this subset of reviews.

5.1.4 Conclusion

The goal of phase 1 was to filter out spam and other unwanted messages from a set of user reviews. This was achieved by launching a job from the Figure Eight platform to reach crowd workers, who could voluntarily participate in this activity. The task that they had to perform required them to classify individual user reviews either as helpful for the developers, or as useless when they had no recognizable value.

Results show that this approach is capable of gathering a crowd, from countries all over the world and through multiple different channels. In total, 177 unique crowd workers originating from 9 different channels contributed to this test between two separate sessions. The automatically deployed contributor survey showed that the contributor generally thought that tasks were set up fairly, were not too difficult to complete and were satisfied with their payment.

In total this test cost $130.80 with a 20% usage fee of the Figure Eight platform included in this price. The job was active for 29 and 82 minutes for each individual session respectively, bringing the total time elapsed to 111 minutes to congregate enough judgments for the classification of all 1,000 user reviews.

The judgments of the contributors were compared with the judgment of the author of the job, and found that the judgments from both parties were in alignment for the majority of the cases. This test showed that the crowd was able to classify the reviews with a precision of 93%, meaning that only 7% of helpful reviews were misjudged as useless by the crowd. Subsequently, depending on the level of strictness, the crowd was able to correctly identify either 74% or 84% of the useless reviews from the dataset. Finally, reviews that received the same judgments from all three different contributors show to be substantially more likely to be classified correctly, compared to reviews that received a judgment where the contributors only partially agreed.

5.2 Second phase

The test for the second phase has a similar setup as for phase 1, only now with the reviews split per sentence instead of entire reviews. Participating crowd workers were instructed to classify these review fragments as helpful or useless. The exact details for the setup of the second test can be viewed in section 4.4.2, but a summary of the setup is shown below in Table 36.

Session	Nr. of Reviews	Required Judgments	Judgment Limit	Nr. Test Questions	Required Passing %	Cost Per Judgment
First	242	726	50	15	70	$0.02
Second	1000	3000	50	15	70	$0.02

Table 36: Summary of the launched job.

The second phase utilizes the output from the first phase and splits them into individual sentences by using an online tool. This action transformed the 508 helpful entire reviews into 1242 sentences. Similar to the first phase, the test for the second phase was conducted in two separate sessions. These sessions were divided into one smaller and one bigger one, with the first session consisting of 242 fragments and the larger one consisting of 1,000 fragments. Each fragment required three separate judgments for classification and the maximum possible judgments that could be contributed per crowd worker was set at fifty. Furthermore, 15 test questions were set up for the eligibility test, which required 70% All these settings are identical to the setup of phase 1, with the exception of the rewards. For phase 2, rewards were set at $0.02 per contribution, which was set to be slightly lower than the rewards for the first phase. This decision was made due to the perceived lower workload for the contributors, as they only had to classify single sentences instead of entire reviews this time around.

5.2.1 Contributor Demographics

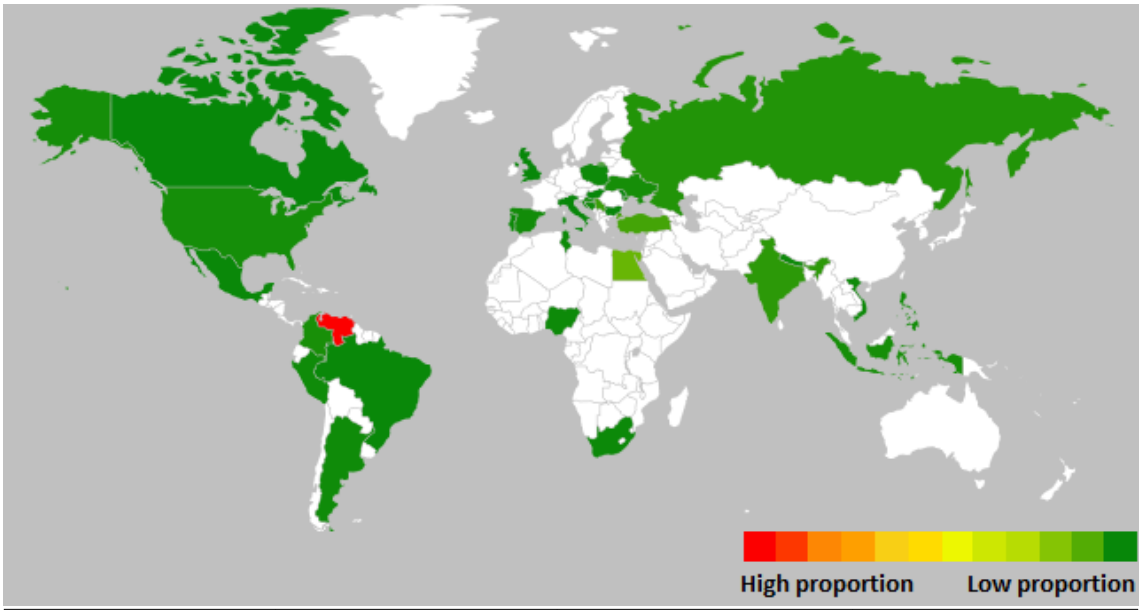
The job was launched with the setup described above and managed to gather a large group of respondents. This is illustrated in Table 37.

Session	**Nr. of Contributors**	**Test Passed**	**Test Failed**	**Average Trust Level**	**Average Contributions**	**Average Testquestions**	**Avg. Incorrect Testquestions**
First	72	63	9	86,43%	18,6	9,4	1,7
Second	209	185	24	86,15%	22,3	9,7	1,7

Table 37: Involved contributors and their performance for both job sessions.

The test for the second phase managed to gather the largest crowd so far. Even though the required work was around 25% more compared to the first phase, a total of 281 crowd workers contributed to the test of phase 2 compared to the 177 of phase 1. Around 90% of the contributors passed the eligibility test, for both the first and the second session. The average trust level of the contributors is largely the same compared to phase 1, but is more consistent between the two sessions as they were almost equal this time around. For both sessions, participants provided around 20 contributions on average. Furthermore, on average only 1,7 test questions were answered incorrectly.

The consistency between the two sessions are quite remarkable when considering that both sessions were launched at two different times of the day. From the perspective of Central European Time (CET), The first session was launched in the morning and the second sessions was launched in the late afternoon. When only looking at Table 37, one could conclude that the time of day does not necessarily impact the the type of participating contributors.



Country	Contributions	Country	Contributions	Country	Contributions	Country	Contributions
Venezuela	3159 (49,3%)	U.S.A.	99 (1,5%)	Nigeria	50 (0,8%)	Bulgaria	10 (0,2%)
Egypt	647 (10,1%)	Philippines	89 (1,4%)	Poland	50 (0,8%)	Canada	10 (0,2%)
Turkey	409 (6,4%)	Croatia	80 (1,2%)	Portugal	50 (0,8%)	Italy	10 (0,2%)
Serbia	340 (5,3%)	Spain	80 (1,2%)	Tunisia	50 (0,8%)	Moldova	10 (0,2%)
India	250 (3,9%)	Marocco	65 (1,0%)	Vietnam	49 (0,8%)	U.K.	10 (0,2%)
Russia	190 (3,0%)	Argentina	60 (0,9%)	South Africa	40 (0,6%)	Hungary	9 (0,1%)
Colombia	120 (1,9%)	Bosnia	60 (0,9%)	Mexico	29 (0,5%)	Kazachstan	9 (0,1%)
Indonesia	110 (1,7%)	Peru	60 (0,9%)	Brazil	20 (0,3%)	-	-
Ukraine	110 (1,7%)	Nepal	50 (0,8%)	Italy	20 (0,3%)	-	-

Table 38: Visualization of the contributions per country.

In proportion, by far the majority of the contributions for the test of the second phase were made by Venezuelans, closing in on the 50% mark. In total 6404 contributions were received to complete the jobs, with 1391 coming from the first session and 5013 coming from the second. The discrepancy in proportion between Venezuela and the other countries is the largest that we have seen so far. This can possibly be attributed to the fact that the second session was launched at a different time of the day compared to all other jobs so far. Furthermore, Posch, Bleier, Flöck, and Strohmaier (2018) consider Venezuelans to be a special case on crowdsourcing platforms due to their current economic situation, which allows them to disproportionately benefit from the exchange rate of the rewarded dollars and their own currency. Nevertheless, plenty of contributions were received from other different countries all around the world as illustrated in the figure accompanying Table 38.

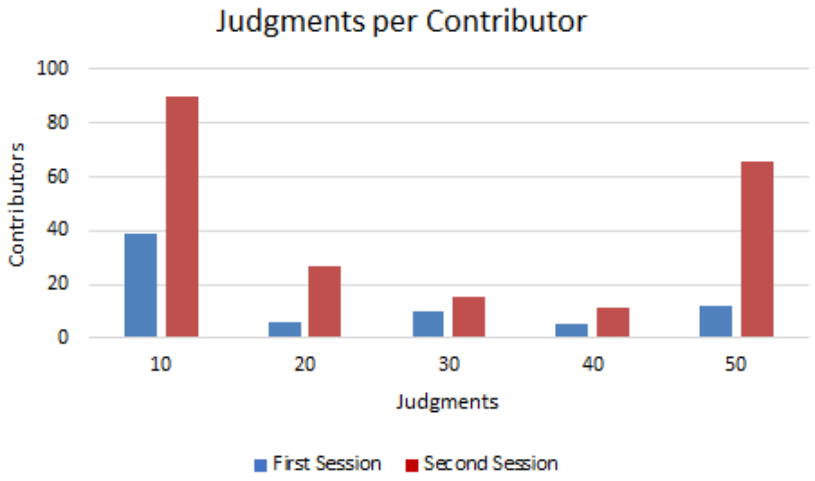


Figure 24: The differences between the amount of contributions made per contributor.

Regarding the received contributions per contributor, Figure 24 shows a similar divide between the crowd workers as in the test for the first phase. The majority of the contributors either contributed either the minimum or the maximum amount of judgments. This divide is most visible for the second session, but is also present for the fist session although to a lesser degree. We assume that the relatively low amount of contributors that reached the maximum of 50 contributions in the first session can be explained by the relative small size of the job. The small size of the job likely caused the work to already run out before contributors were able to reach the maximum, even when contributors intended to reach said maximum. Nevertheless, Figure 24 illustrated that it is possible to keep crowd workers interested for extended periods of time for this kind of work.

	First Session (242 Reviews		**Second Session (1000 Reviews)**	
Channels	**Contributors**	**Avg. Trust Level**	**Contributors**	**Avg. Trust Level**
FigureEight	31	86%	89	83%
NeoBux	8	88%	106	84%
ClixSense	43	82%	53	87%
InstaGC	1	100%	2	85%
GetPaid	1	100%	1	92%
GiftHuntersClub	1	80%	-	-
KeepRewarding	1	70%	-	-
BitcoinGet	-	-	1	100%
Overall	**86**	**84%**	**252**	**85%**

Table 39: Source channels of the contributions and the average trust levels of their contributors.

The three channels that were responsible for providing the majority of the contributors in the first phase are again the biggest suppliers of contributors for the first phase. We can assume that in general, crowd workers are the most active on NeoBux, ClixSense and Figure Eight itself. However, a big discrepancy can be observed between the two sessions for NeoBux. For the first session, only 8 out of 86 contributors originated from NeoBux. This increased to 106 out of 252 in the second session and therefore became the biggest supplier of crowd workers in said session. No major deviations could be observed regarding the trust levels of the contributors between the two sessions. However, the overall trust level over the entire test of phase 2 is slightly lower than of the test for phase 1.

5.2.2 Job Statistics

The automatically deployed contributor survey at the end of the job allows contributors to voluntarily provide feedback on their perception of the work. The results of this survey for the two sessions in phase 2 is shown in Table 40.

Session	**Respondents**	**Instruction Clarity**	**Test Question Fairness**	**Ease of Job**	**Pay**	**Overall**
First	14	3.9 / 5	3.3 / 5	3.5 / 5	3.7 / 5	**3.7 / 5**
Second	24	4.1 / 5	3.9 / 5	3.5 / 5	3.8 / 5	**3.7 / 5**

Table 40: Contributor perception of the tasks.

While the survey received a relatively low amount of respondents for both sessions, the results are still quite consistent between them. With an overall rating of 3,7 out of 5, contributors were slightly less satisfied with the work compared to phase 1 illustrated by a drop in rating of 0,5. This can be explained by the lower rating for the ease of the job, illustrating that the work was harder to complete than the required work in phase 1. This was somewhat expected, as it is overall harder to interpret single sentences without the context of the entire review. Nevertheless, the overall scores show that contributors still liked the work, thought that the instructions were quite clear and that the pay was fair even with the reduction in payment compared to phase 1.

Session	**Date**	**Launch**	**Finished**	**Duration**	**Review Judgments**	**Untrusted Judgments**	**Test Question Judgments**	**Total Judgments**	**Total Cost**
First	23/05/2019	11:21	11:49	28 min.	726	72	665	1463	$21.36
Second	29/05/2019	16:40	17:24	44 Min.	3000	159	2091	5250	$84.96

Table 41: Launch time and completion statistics for the jobs in this phase.

This reduction in pay is illustrated in the total costs for the jobs in the two sessions as shown in Table 41. Although the jobs that were launched for the second phase were the largest so far, the total cost came down to $106.32 with the 20% usage fee of the platform included which is lower than the total costs for all tests in phase 1. Furthermore, the total time required to finish the jobs from both sessions is significantly lower than for the first phase. The second session alone in this phase had an equal work load to the entirety of phase 1, but only took 44 minutes to reach completion compared to the total time of 111 minutes that were required to finalize both sessions within phase 1. This effect could be attributed to the fact that session two was launched at a different time of the day compared to all previously launched jobs, conveying the impression that the crowdsourcing channels are more active at this particular time of day. This effect caused that the largest job launched so far was completed in the least amount of time relative to the job size.

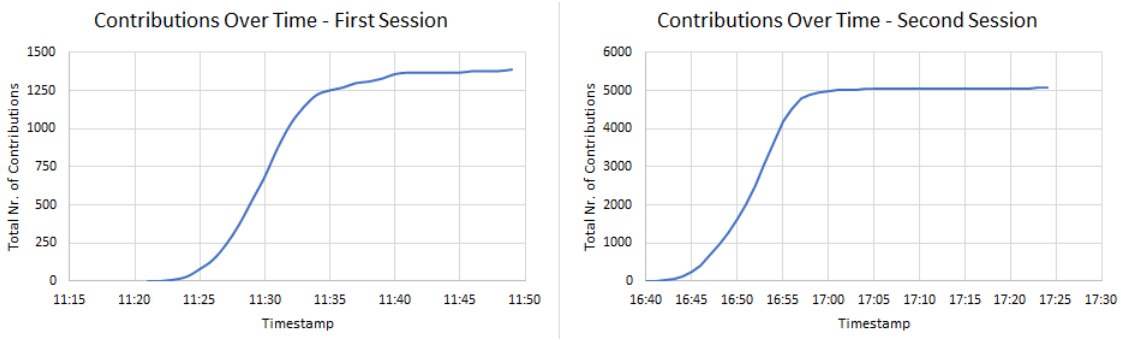


Figure 25: Distribution of received contributions over time.

The course of the sessions over time show similar trends that were previously observed during the tests of phase 1. The total received contributions over time as depicted in Figure 25. For both sessions, the majority of the contributions were already received at the halfway point. The remaining time to finalize the job depends on the contributions that have to complete the already distributed work. This remaining time is depicted in the long tails at the end of both graphs depicted in Figure 25. This tail is especially long for the second session, as the total number already nearly reaches the 5,000 mark while the job still requires an additional 25 minutes to finalize.

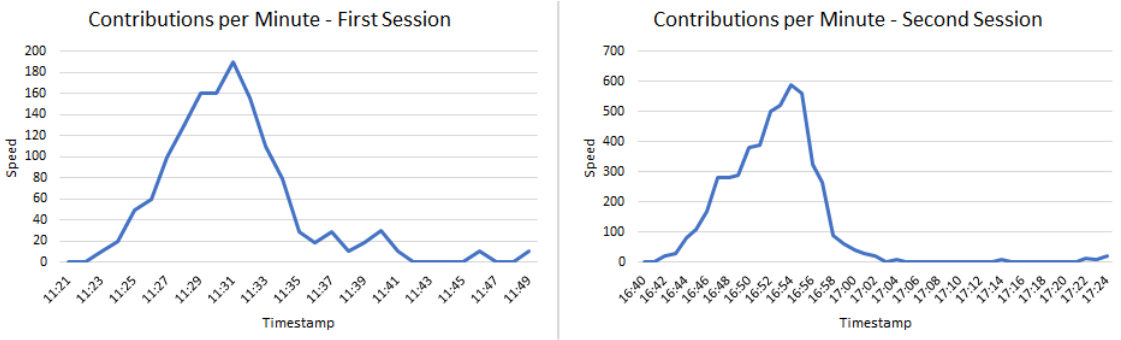


Figure 26: Contributions received per minute for both sessions.

The graphs depicted in Figure 26 provide additional information regarding the course of both sessions. In the first minutes of each session, the contributions received per minute slowly starts ramping up. When nearing the halftime point, a big drop-off can be observed, further substantiating the notion that the work running out is responsible for the tails observed in Figure 25. The number of contributions received per minute also does not have appeared to stabilize before the drop-off occurred, conveying the impression that the potential maximum speed was never reached.

Session	Average Speed	σ Speed	Average Time per Contributor	σ Average Time
First	42 Judg. per Min.	58,2	12,3 Seconds per Judgment	17,5
Second	113 Judg. per Min.	177,3	10,0 Seconds per Judgment	10,9

Table 42: Average speed of the job and average time per judgment per contributor

The average speeds shown in Table 54 further substantiate the conclusion that the second session was completed by far the quickest relative to its size. The average of 113 judgments received per minute is around 2,5 times higher than the average speeds we have seen for any session launched so far. With 42 judgments received on average, the speed of the first session is on par with the two sessions from the first phase. The relatively high standard deviation of the speed for both sessions however indicate that these averages fluctuated quite a lot over the entire duration of the sessions.

For the average times per contributors to submit an individual judgment, Overall, the average times per contributor per judgment are quicker than those of the first session. Compared to the first phase, judgments per contributor were quicker on average during the second phase tests. This was expected as phase 2 only required the workers to analyze single sentences, compared to the entire reviews in phase 1.

5.2.3 Outcome

This section explores the differences and similarities of the classifications by the crowd and the researcher by comparing them. It therefore presents the quality of the results, both before and after the review process by the researcher, which is described in detail in section 4.5.

	Researcher		Crowd		Researcher		Crowd	
Session	**Helpful #**	**Useless #**	**Helpful #**	**Useless #**	**Helpful %**	**Useless %**	**Helpful %**	**Useless %**
First	110	132	121	121	47,4%	52,6%	50%	50%
Second	453	547	562	438	45,3%	54,7%	56,2%	43,8%
Overall	563	679	683	559	45,3%	54,7%	55%	45%

Table 43: Comparison of helpful and useless reviews identified by the researcher and the crowd.

The initial classification of the review fragments was relatively close between the crowd and the researcher. The classification of the researcher resulted in a slight majority in the group of useless review fragments, while the crowd were more likely to mark them as helpful. The classification statistics inbetween the two sessions are quite consistent, even though reviews from the second session were a bit more likely to be marked as helpful by the crowd.

Session	**True Positives**	**Actual TP**	**True Negatives**	**Actual TN**	**False Positives**	**False Negatives**	**Precision**	**Recall**
First	100	132	89	110	21	32	0,83	0,76
Second	378	547	393	453	60	169	0,86	0,69
Total	478	679	482	563	81	201	0,86	0,70

Table 44: Detailed comparison of the results of the crowd before the review.

The precision and recall values depicted in Table 56 illustrate how accurate the classification of the crowd was in respect to the classification of the researcher. On average, the precision value illustrates that 86% of the review fragments that were marked as useless by the crowd, were marked correctly. In total, the crowd was able to identify 70% of all useless review fragments in the dataset, meaning that 30% of the useless fragments would remain in the dataset. However, these values represent the accuracy of the crowd before the answer reviewing process.

During the reviewing process, the review fragments that did not have a match between the classification of the researcher and the crowd were revisited. During the review, the classification of the reviewer was corrected based on the reasonable expectations on the crowd using only the information provided to them. In other words, the classifications that required significantly more knowledge than was made available for the crowd were corrected. The purpose of this process was to compensate for the knowledge that the researcher possessed about the products that the reviews originated from and to compensate for any bias that the researcher acquired when the entire reviews were read for the classification of the previous phase. No compensation was provided for errors that were made due to the lack of context, as that is inherent to the nature of the work in this phase. The effects of this reviewing process and the implications for the previously discussed results will be discussed below.

	Researcher		Crowd		Researcher		Crowd	
Session	**Helpful #**	**Useless #**	**Helpful #**	**Useless #**	**Helpful %**	**Useless %**	**Helpful %**	**Useless %**
First	124	118	121	121	51,2%	48,7%	50%	50%
Second	509	491	562	438	50,9%	49,1%	56,2%	43,8%
Overall	633	609	683	559	51%	49%	55%	45%

Table 45: Comparison of helpful and useless reviews by both parties after the reviewing process.

Table 45 illustrate the shift in the classification of the researcher after the reviewing process. During the reviewing process, the researcher had to concede that some reviews had to be marked as helpful when using only the constructed guidelines provided to the crowd. This resulted in an increase of 70 review fragments in the helpful category, bringing the total helpful reviews by the reviews to 633. This brings the proportion of helpful reviews much closer to the classification of the crowd to a discrepancy of only 4%.

Session	**True Positives**	**Actual TP**	**True Negatives**	**Actual TN**	**False Positives**	**False Negatives**	**Precision**	**Recall**
First	103	118	106	124	18	15	0,85	0,87
Second	390	491	460	509	49	102	0,89	0,79
Total	**493 (+15)**	**609 (-70)**	**566 (+84)**	**633 (+70)**	**66 (-15)**	**117 (-84)**	**0,88 (+0,02)**	**0,81 (+0,11)**

Table 46: Detailed comparison of the results of the crowd after the review.

The implications that the review process had on the accuracy of the performance of the crowd is quite substantial. The recall value increased to 0,81 which means that the crowd was able to identify 81% of all useless reviews within the dataset. For precision, only a minor increase could be observed, meaning that the crowd classified 88% of the useless reviews correctly as useless. However, this does mean that 12% of the fragments that were marked as useless actually contain helpful information according to the researcher, meaning that they would be incorrectly discarded. Furthermore, the new recall value means that 19% of all useless review fragments remain in the dataset and were unfortunately not filtered out.

Session	**Reviews**	**Unanimous Judgments**	**Correct Judgments**	**Incorrect Judgments**	**Indifferent Judgments**	**Correct Judgments**	**Incorrect Judgments**
First	242	152	140	12	90	63	27
Second	1000	611	527	84	389	244	145
Total (%)	**1242 (100%)**	**763 (61,4%)**	**667 (87,4%)**	**96 (12,6%)**	**479 (38,6%)**	**307 (64,1%)**	**172 (35,9%)**

Table 47: Comparison between full agreement and indifferent judgments.

Differences in accuracy of the crowd could also be observed between fragments that were classified unanimously by all three contributors and fragments where the contributors were indifferent and did not reach a full agreement. In 61,4% of all cases, three contributors reached a unanimous agreement separately from each other. In 87,4% of those cases, the contributors reached the correct answer. In contrast, only in 12,6% of these cases, all three contributors agreed on the wrong answer. No unanimous agreement could be reached in 38,6% of all cases and 64,1% of these cases were still correct. This is lower than when contributors reached a full agreement, but is still quite substantial. No unanimous agreement could therefore be reason for caution, but in no substantiated way could mean that these contributions should not be trusted.

5.2.4 Conclusion

Phase 2 of the method has as its goal to filter out as many useless review fragments as possible. To achieve this, the output of the first phase was automatically split per sentence, resulting in a dataset of 1242 review fragments which were used as input for the second phase. A task description catered to individual review fragments was construct and this was deployed on Figure Eight. Using similar job settings as in the first phase, crowd workers were instructed to classify these fragments as either helpful for the developers or useless when they had no recognizable value.

As previously demonstrated, this particular setup was able to gather a large group of contributors originating from countries all over the world. These contributors were gathered through multiple different channels, all similar to the results of the first phase of the method. The automatically deployed contributor survey at the end of the tasks showed that the crowd workers

were overall satisfied with the work, thought that the instructions were clear and the eligibility criteria and payments were fair. However, overall ratings decreased a little compared to the results from phase 1. This can most likely be attributed to the more difficult nature of the task, as the fragments provide significantly less context compared to entire reviews. Nevertheless, the entire classification of all 1242 took a mere 72 minutes in total, indicating that crowd workers are still very eager to participate in this kind of work.

The total cost to reach full completion of the required work for phase 2 came down to $106.32 with the 20% usage fee of the platform included. This was cheaper that the total costs for the tests of phase 1, even though the absolute number of required judgments were higher. This was caused by the lowered rewards for individual judgments from three to two cents. This decision was made due to the decreased required effort to complete the tasks by the contributors, as they only had to read single sentences instead of entire reviews. No clear and direct effects of the lower rewards could be observed, as this phase has gathered the largest crowd and was completed in the least amount of time so far.

The quality of the classification itself was slightly lower compared to the results of the first phase when comparing the results from the crowd with the classification of the researcher. The tests for phase 2 showed that the crowd was able to classify useless results with a precision of 88%, meaning that 12% were incorrectly discarded helpful fragments. Overall, the crowd was able to identify 81% of all total useless fragments from the entire dataset. This is a substantial reduction, although it also means that still 19% of useless fragments remain in the dataset. Detailed analysis of the wrongly classified fragments showed that the crowd had difficulties dealing with sentences calling for a generic fix and had trouble with dealing with the concept of ease of use. These aspects will have to be considered during following phases of the method.

5.3 Third Phase

The third phase of the method entailed the classification of helpful review fragments into five different categories. The classification of these fragments required the separate judgments of six individual contributors. This reduced the chances of a stalemate occurring, as this ensured that at least one of the five categories would have be chosen twice. In other cases of stalemates, the answers of the contributors with the highest trust levels will be chosen.

In contrast to the test of the previous two phases, the third phase was executed in one single job session. More details about the exact setup of the test for the third phase can be found in section 4.3. A summary of the configuration of the test for the third phase is shown in Table 48.

Session	Nr. of Reviews	Required Judgments	Judgment Limit	Nr. Test Questions	Required Passing %	Cost Per Judgment
First	683	4098	50	15	70	$0.02

Table 48: Summary of the launched job.

Phase 3 uses the output of phase 2 and aims to further process them until they can be categorized into their respective categories. The input for this particular test are therefore the 683 helpful review fragments that resulted from the tests for phase 2. Due to the six required judgments per review fragments, a minimum of 4098 judgments will be required to finalize the job. The settings for the judgment limit per contributor and the settings for the eligibility test were kept equal to all previous test for all previous phases. However, the pay was kept relatively lower under the recommended minimum hourly wage compared to the required workload of this job due to budget constraints. This may result in the job taking a bit longer than it should to complete as it is likely that less workers are willing to do the work. This means that the time to reach completion could not necessarily be directly compared to the times of the previous two phases.

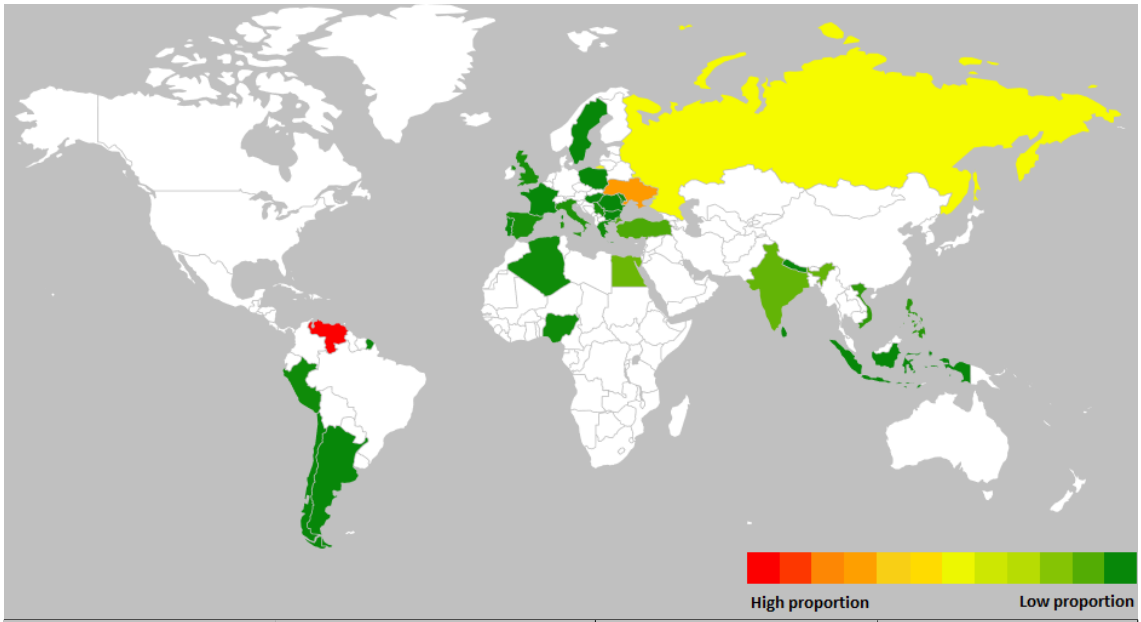
5.3.1 Contributor Demographics

Utilizing the settings described above, the job was launched as a single session containing all 683 helpful review settings. A summary of the involved contributors is provided in Table 49.

Session	**Nr. of Contributors**	**Test Passed**	**Test Failed**	**Average Trust Level**	**Average Contributions**	**Average Testquestions**	**Avg. Incorrect Testquestions**
First	226	164	62	75,4%	31,8	12,4	2,8

Table 49: Involved contributors and their performance for the single job session.

First and foremost, the single job session for the first phase managed to gather the largest crowd for a single session for far. To reach completion, 226 contributors were involved, of which 62 were denied further participation as they did not pass the eligibility test. With 27% of contributors being denied further access, this is in turn the largest proportion of contributors that failed the eligibility test. This can be viewed as the first indication that tasks for the first phase are harder and require more knowledge compared to the tasks in the preceding phases. Furthermore, the average trust levels of the participating contributors is the lowest of all performed tests, indicating that the quality of the results generated by this crowd could potentially be lower. On average, they provided nearly 32 contributions per contributor. Contributors also encountered on average 12,4 test questions, which is close to expectations with an average of around 30 contributions per contributor. Out of these test questions, 2,8 were answered incorrectly on average which is the highest margin observed over all test sessions.



Country	Contributions	Country	Contributions	Country	Contributions	Country	Contributions
Venezuela	63 (27,9%)	Italy	4 (1,8%)	Algeria	2 (0,9%)	Nepal	1 (0,4%)
Ukraine	45 (19,9%)	United Kingdom	3 (1,3%)	Kazachtstan	2 (0,9%)	Indonesia	1 (0,4%)
Russia	32 (14,2%)	Spain	3 (1,3%)	Chile	1 (0,4%)	Poland	1 (0,4%)
Egypt	14 (6,2%)	Serbia	3 (1,3%)	Romania	1 (0,4%)	Sri Lanka	1 (0,4%)
India	13 (5,8%)	Greece	2 (0,9%)	Bulgaria	1 (0,4%)	France	1 (0,4%)
Turkey	10 (4,4%)	Nigeria	2 (0,9%)	Sweden	1 (0,4%)	Hungary	1 (0,4%)
Vietnam	6 (2,7%)	Peru	2 (0,9%)	Argentina	1 (0,4%)	Marocco	1 (0,4%)
Philippines	5 (2,2%)	Moldova	2 (0,9%)	Portugal	1 (0,4%)	-	-

Table 50: Visualization of the contributions per country.

The origin of the contributors a bit more evenly distributed among workers over the globe compared to the previous phases. Venezuelans are still the largest group, but countries such as Ukraine and Russia are much closer compared to the largest group. As the time of day the lower proportion of Venezuelans could be attributed to the more difficult nature of the tasks and the larger proportion of people that failed the eligibility test. No decisive evidence could be found for this claim however.

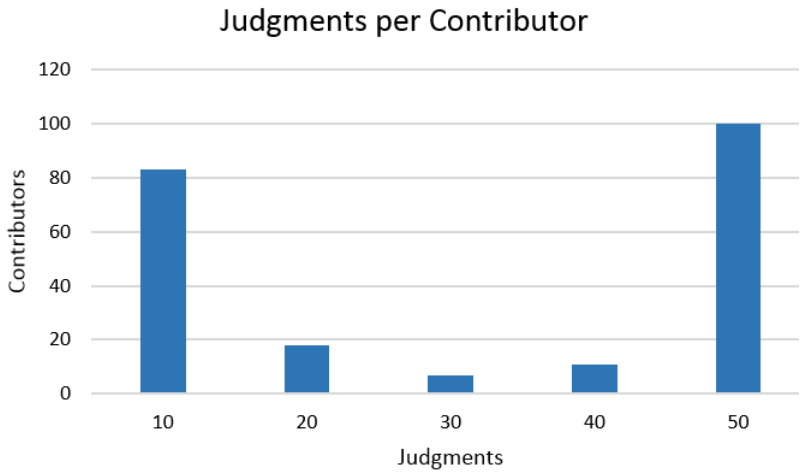


Figure 27: The differences between the number of contributions made per contributor.

Regarding the individual contributors per contributor, a similar divide can be observed as in the previous two phases as illustrated in Figure 27. This further substantiates the conclusion that crowd workers tend to gravitate towards doing either the minimum or maximum number of allowed tasks. Same distribution as before. However, this time around the group that maxed out on the maximum allowed amount of contributions is the largest group. Apparently, contributors that passed the illegibility test were committed to contribute as much as they were allowed.

Channels	Contributors	Avg. Trust Level
ClixSense	98	79%
FigureEight	68	77%
NeoBux	51	68%
InstaGC	2	60%
SuperRewards	2	75%
GetPaid	2	55%
GiftHuntersClub	1	86%
Wannads	1	64%
KeepRewarding	1	79%
Overall	**226**	**71%**

Table 51: Source channels of the contributions and the average trust levels of their contributors.

The notion that the three main channels that supply crowd workers are Figure Eight, NeoBux and ClixSense is further substantiated from the results gathered for the third phase, as shown in Table 51. We can therefore conclude that crowd workers are the most active on NeoBux, ClixSense and Figure Eight itself. The average trust levels of all contributors involved in the third phase has been the lowest that we have seen between the three phases. NeoBux is the most impactful channel for this development, as nearly a quarter of all contributors were provided through NeoBux when contributors only had an 68% trust rating on average.

5.3.2 Job Statistics

The automatically deployed contributor survey at the end of the job allows contributors to voluntarily provide feedback on their perception of the work. A summary of the results gathered from the respondents during for phase three is provided in Table 52.

Session	Respondents	Instruction Clarity	Test Question Fairness	Ease of Job	Pay	Overall
First	34	3.8 / 5	3.7 / 5	3.5 / 5	3.7 / 5	**3.7 / 5**

Table 52: Contributor perception of the tasks.

Even though the complexity of the tasks was much more advanced compared to the previous phases, similar satisfaction scores were reached as in phase 2. The overall rating of 3.7 out of 5 is identical to the overall rating of phase 2. We expected a drop in satisfaction regarding the payments, but respondents of the survey rated the payment satisfaction with a 3.7 out of 5, also identical to the rating from phase 2. Furthermore, the crowd workers again thought that the provided instructions were adequately clear and that the test questions were fair in general. Even though we expected the work to be more demanding and therefore less popular for the crowd workers, the respondents of the survey do not necessarily agree with this expectation.

Session	Date	Launch	Finished	Duration	Review Judgments	Untrusted Judgments	Test Question Judgments	Total Judgments	Total Cost
First	13/06/2019	11:04	13:24	140 min.	4098	270	1943	6311	$117.60

Table 53: Launch time and completion statistics for the jobs in this phase.

The job itself was launched late in the morning and completed on the same day. The time to reach completion was 140 minutes, which is by far the longest run time of a single job between all three phases. In this time, 4098 judgments were acquired for the classification of the 683 review fragments. Furthermore, 1943 additional judgments were gathered on the test questions for the completion of all eligibility tests for all participating crowd workers. The number judgments for the eligibility test is relatively high, which can be attributed to the fact that this particular job had the largest percentage of people who failed to pass the initial eligibility test. Additionally, 270 untrusted judgments were received by workers who passed the eligibility test, but failed to perform adequately on the remaining quality control questions. Therefore, a grand total of 6311 judgments were required to reach job completion, which brought the entire cost of the job to a total of $117.60.

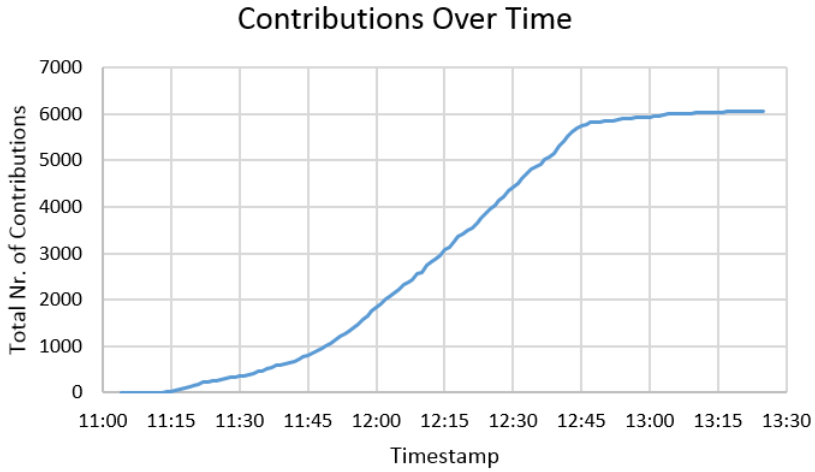


Figure 28: Distribution of received contributions over time.

The course of the entire job is depicted in Figure 28, showing the total number of contributions received over the entire job duration. While the curve is similar to what was observed in the previous two phases, the finalization phase represented by the tail and the end of the curve is smaller relative to the entire graph. For the test in phase 3, this finalization phase were the available work started running out started after two thirds of the entire job duration, compared to the previously observed halfway point. Nevertheless, the graph depicted in Figure 28 shows that the vast majority of the required judgments was reached after 1 hour and 45 minutes. The remaining 35 minutes were required by the crowd workers to finalize the remaining tasks that were already distributed.

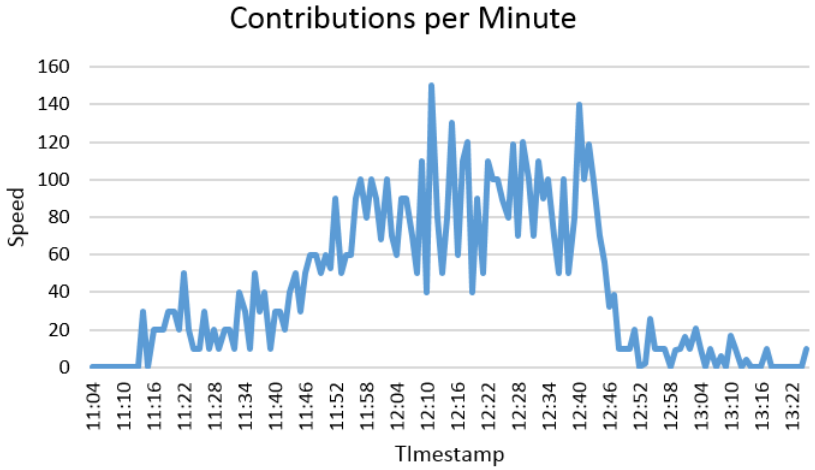


Figure 29: Contributions received per minute.

Similar to all previously launched jobs, the contributions received per minute fluctuated quite a lot over the entire duration of the job, as shown in Figure 29. The contributions received per minute quickly started ramping up after job launch before it somewhat stabilized in the middle section. Afterwards, again a quick drop off can be observed at the point where the work started running out and the finalization phase commenced. The stabilization of the graph in the middle section indicates that the maximum potential speed of the job was reached of around 100 contributions received per minute with this kind of configuration.

Session	Average Speed	σ Speed	Average Time per Contributor	σ Average Time
First	43 Judg. per Min.	39,6	23,6 Seconds per Judgment	23,4

Table 54: Average speed of the job and average time per judgment per contributor

The average speed over the entire duration of the job however is a bit lower. During the 140 minutes that the job took to reach completion, 43 judgments were received on average per minute.

As previously indicated and further substantiated with the standard deviation of this average speed, these speeds fluctuated quite a lot over the entire duration of the job, although these fluctuations are less extreme compared to the tests in previous phases. Furthermore, it took the contributors on average 23,6 to provide a single judgment, which is around twice as long compared to the previous phases. We attribute this to the overall complexity of the job and the more advanced and longer job description that was provided to the contributors.

5.3.3 Outcome

This section explores the differences and similarities of the classifications by the crowd and the researcher by comparing them. It therefore presents the quality of the results, both before and after the review process by the researcher, which is described in detail in section 4.5.

	Absolute Values		**Proportion of Total**	
Category	**Researcher**	**Crowd**	**Researcher**	**Crowd**
None	136	92	19,9%	13,5%
Feature Request	100	102	14,6%	14,9%
Stability Feedback	120	182	17,6%	26,6%
Performance Feedback	52	60	7,6%	8,8%
Quality Feedback	275	247	40,3%	36,2%
Total	**683**	**683**	**100%**	**100%**

Table 55: Comparison between the classifications of the researcher and the crowd.

Table 55 shows the distribution of frequency of the chosen categories between the crowd and the researcher. While most of the categories are somewhat close in the absolute value sense between the two parts, preferences for different categories can also be observed. For instance, the researcher marked a substantial amount of fragments to belong in the *None* category, while the crowd classified a substantial amount of fragments more in the *Stability* category. Other deviations can be observed in the *Quality* category, although to a lesser degree. For the *Feature* and the *Performance* category, the classification is much closer between both parties in the absolute sense.

Crowd Judgments

Researcher	**Category**	**None**	**Feature**	**Stability**	**Performance**	**Quality**
	None	61	11	17	5	42
	Feature	4	77	10	5	4
	Stability	4	2	107	3	4
	Performance	3	2	9	23	15
	Quality	20	10	39	24	182

Figure 30: Confusion Matrix comparing the answers of the crowd with the answers of the researcher.

To compare the accuracy of the classification of the crowd compared to the classification of the researcher, a confusion matrix was constructed as shown in Figure 30. This confusion matrix allows us to identify where the most errors were made and between which two categories the crowd workers were most confused. The majority of the mistakes are between the *None* and the *Quality* category in both ways. Where the crowd researcher selected *Quality*, the crowd selected *None* in 20 instances and where the researcher chose *None*, the crowd selected *Quality* in 42 instances. Other areas of errors can be observed between the *Performance* and the *Quality* category and the *Stability* and the *Quality* categories, although to a lesser degree.

Category	**True Positives**	**Actual TP**	**True Negatives**	**Actual TN**	**False Positives**	**False Negatives**	**Precision**	**Recall**
None	61	136	516	92	31	75	0,44	0,66
Feature	77	100	558	102	25	23	0,77	0,75
Stability	107	120	488	182	75	13	0,89	0,59
Performance	23	52	594	60	37	29	0,44	0,38
Quality	182	275	343	247	65	93	0,66	0,74
Total	450	683	-	-	-	-	0,64	0,62

Table 56: Detailed comparison of the results of the crowd before the review.

When analyzing the specific precision and recall values for each individual category, it becomes evident that the crowd was the most accurate in classifying the fragments as *Stability* or as a *Feature request*. The worst performing category is the *Performance* category, but those values may be skewed a bit as this was by far the smallest category compared to the total amount of fragment out of the five categories. The total precision score means that on average 64% of the classification by the crowd matched the classification of the researcher. However, as these values do not take into account the different weights of the categories, the micro average of the total precision was also calculated and came down to 66%. This means that two thirds of the classification of the crowd matched the classification of the researcher before the reviewing process.

At this point, the reviewing process commenced which is described in detail in section 4.5. For this particular reviewing process, answers were corrected when a review mentioned only a tiny aspect of the chosen category, even if it was not the predominant one. As it was possible that a review fragment considered multiple categories, this compensation was necessary as classifications would be marked as incorrect when the researcher and the crowd disagreed on the importance of said aspect. As the main objective of this test was to see whether the crowd was capable in recognizing aspects related to the five categories, answers that fit these criteria would be marked as correct from this point onwards.

	Absolute Amount		**Proportion of Total**	
Category	**Researcher**	**Crowd**	**Researcher**	**Crowd**
None	117 (-19)	92	17,1% (-2,7%)	13,5%
Feature Request	113 (+13)	102	16,5% (+1,9%)	14,9%
Stability Feedback	144 (+24)	182	21,1% (+3,5%)	26,6%
Performance Feedback	46 (-6)	60	6,7% (-0,9%)	8,8%
Quality Feedback	263 (-12)	247	38,5% (-1,8%)	36,2%
Total	**683**	**683**	**100%**	**100%**

Table 57: Comparison between the classifications of the researcher and the crowd after the review.

With a maximum shift of 3,5% for the *Stability* category, the reviewing process did not have a large impact in the proportions of selected categories, as illustrated in Table 57. The researcher had to concede that the *None*, *Performance* and *Quality* categories were chosen a few instances too many. On the other hand, the reviewing process showed that the *Feature* and *Stability* category would have also been a correct classification in a couple of instances. In general however, no large implications of the reviewing process could be derived from this data.

Crowd Judgments

Researcher	**Category**	**None**	**Feature**	**Stability**	**Performance**	**Quality**
	None	67	4	14	4	28
	Feature	5	94	8	5	1
	Stability	3	1	134	3	3
	Performance	3	1	6	29	7
	Quality	14	2	20	19	208

Figure 31: The confusion matrix after the review.

As depicted in the confusion matrix shown in Figure 31, the overall number of errors were reduced by the reviewing process. However, still the largest area of errors can be observed between the *None* and *Quality* categories, although to a lesser degree. The reviewing process also made more evident that errors were made between the *Performance* and the *Quality* categories. Additionally, the crowd also disproportionally went for the *Stability* category, as the researcher preferred the *None* or the *Quality* category in 34 of those instances. Furthermore the confusion matrix shows that potentially irrelevant reviews belonging to the *None* category are most likely to end up in the *Quality* category with this job configuration.

Category	True Positives	Actual TP	True Negatives	Actual TN	False Positives	False Negatives	Precision	Recall
None	67	117	541	92	25	50	0,57 (+0,13)	0,73 (+0,07)
Feature	94	113	468	102	8	19	0,83 (+0,06)	0,92 (+0,17)
Stability	134	144	388	168	48	10	0,93 (+0,04)	0,80 (+0,21)
Performance	29	46	493	70	31	17	0,63 (+0,19)	0,41 (+0,03)
Quality	208	259	292	247	39	55	0,80 (+0,14)	0,84 (+0,10)
Total	562	683	-	-	-	-	0,75 (+0,09)	0,74 (+0,12)

Table 58: Detailed comparison of the results of the crowd after the review.

The reviewing process did have a relatively large impact on the precision and recall values for each of the five categories. Large increases in precision can be observed for the *Performance* and the *Quality* category, meaning that the review fragments marked as such were correct in more cases than originally assumed. Furthermore, the recall value for the *Stability* category increase quite substantially as well, indicating that the crowd was able to identify more stability related feedback out of the entire dataset than originally assumed. Overall, the average macro precision of the crowd over all five categories increased to 75%, meaning that three quarters of the review fragments were classified in accordance with the classification of the researcher. When taking the different class weights into account, this precision value increased to 79%.

Agreement	Frequency	Correct	Incorrect	Accuracy
Six out of six	85 (12%)	85	0	100%
Five out of six	144 (21%)	131	13	91%
Four out of six	170 (25%)	145	25	85%
Three out of six	196 (29%)	128	68	65%
Two out of six	88 (13%)	43	45	49%
Total	**683 (100%)**	**532**	**151**	**78%**

Table 59: Accuracy of the aggregated answer for the different levels of agreement between the six contributors.

Concludingly, the different levels of agreement between the six contributors per fragment was analyzed as well. As shown in Table 59, the degree to which the classification of the researcher and the classification of the crowd matched with each other depended quite heavily on the different levels of agreement between the contributors. When all six out of six contributors agreed, the classification matched the full 100% between the two parties. This occurred in 85 out of the 683 instances. This accuracy rating diminished to 91% and 85% for the agreement ratings of five out of six and four out of six respectively. The accuracy rating dipped below the average of 78% for the instances where only three or two of the contributors agreed on a single category. These results warrant the conclusion that the quality of the results will scale with the number of involved contributors per review fragment. With only six contributors, it was possible to reach a 100% with the classification of the researcher. This is likely to increase in frequency even more with either the involvement of more contributors, or by keeping the judgment phase active until at least six contributors agreed on a single category. Both these actions will increase the cost of the phase, but are likely to improve the quality of the results significantly.

5.3.4 Conclusion

Phase 3 entailed the classification of 683 review fragments into 5 different requirements categories. The goal of this phase was to reach a classification by the crowd that aligned as much a possible with the classification of the researcher.

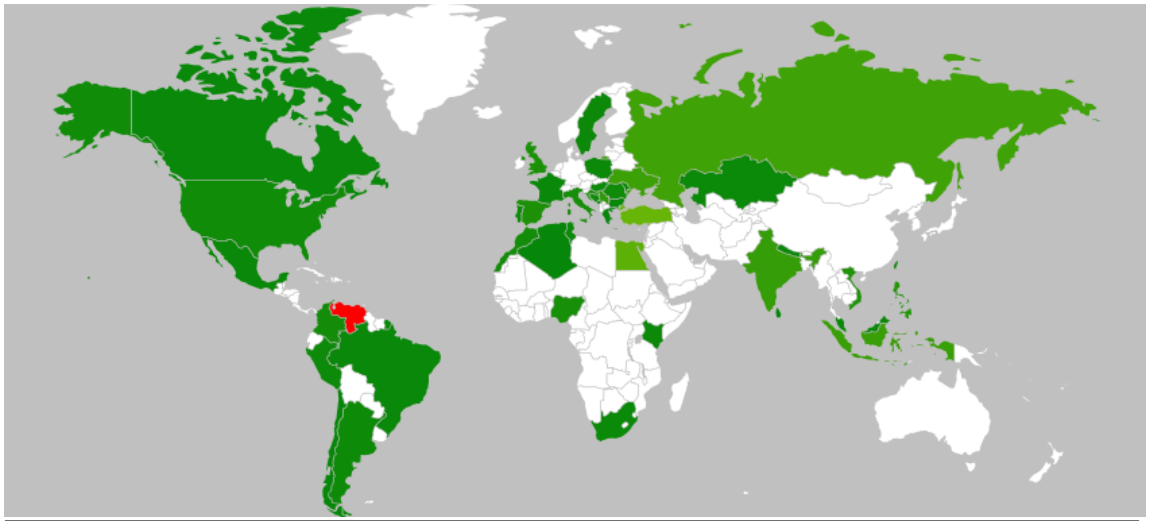
To reach this goal, one single job was launched that involved 226 crowd workers in the classification process of these fragments. It took the entire group of crowd workers a combined total of 140 minutes to reach all required judgments (six per fragment) to complete the job. The automatically distributed contributor survey at the end of the tasks showed that the crowd workers were adequately satisfied with the work and gave a rating of 3.7 out of 5, even with the rewards being set below the recommended amount due to budget constraints. Due to the lower offered rewards, the total cost of the job came down to a total of $117.60 which suppressed the total cost of the entire method, but also likely negatively affected the required time to reach completion.

When comparing the performance of the classification of the crowd with the golden standard created by the researcher, a matching percentage of around 78% was discovered after the answer reviewing process. Some misalignment between categories was also identified, mainly between the *None* and *Quality* categories, and to a lesser degree the *Performance* as well. Precision values were shown to be the lowest for the *Performance* category as well, but these results appear to be a bit skewed due to the relatively smaller size of this particular category compared with the others. Point of improvement were identified for the task description, as it did not account for the difference between stability of the product that the review was for originally and connections with third parties. Furthermore, stability related feedback can also be provided in the form of comments that specify when something in fact did work correctly. Lastly, when reviewers mentioned something that they could not do in the app, it was often marked as a stability issue instead of a feature request by the contributors.

Concludingly, the agreement rating between the six contributors per review fragment was found to have a significant aspect on the quality of the classification. When all six out of six contributors agreed on a single category, a 100% matching rate was identified with the classification of the researcher. This unfortunately diminished to 45% when only two out of six contributors agreed on a single category.

6 Performance Overview

This chapter is a brief summary of the performance of the constructed method in its entirety. It aggregates the results generated from all different tests for all three phases that were tested. This will achieve both a clearer view of the workings of the method in its entirety and of the generated results.



Country	Contributions	Country	Contributions	Country	Contributions	Country	Contributions	Country	Contributions
Venezuela	4629 (40,1%)	Vietnam	245 (2,1%)	Romania	91 (0,8%)	Sweden	51 (0,4%)	Brazil	20 (0,2%)
Turkey	931 (8,1%)	Spain	203 (1,8%)	Taiwan	90 (0,8%)	Kenya	50 (0,4%)	Canada	20 (0,2%)
Egypt	821 (7,1%)	United Kingdom	203 (1,8%)	Argentina	71 (0,6%)	Montenegro	50 (0,4%)	Moldova	12 (0,1%)
Russia	542 (4,7%)	Croatia	180 (1,6%)	Portugal	64 (0,6%)	Tunisia	50 (0,4%)	Bulgaria	11 (0,1%)
Ukraine	514 (4,5%)	Nigeria	161 (1,4%)	Peru	62 (0,5%)	Kazakhstan	43 (0,4%)	Algeria	2 (0,02%)
Indonesia	490 (4,2%)	Italy	144 (1,2%)	Bosnia Herzegovina	60 (0,5%)	South Africa	40 (0,3%)	Greece	2 (0,02%)
India	433 (3,8%)	Colombia	120 (1,0%)	Hungary	60 (0,5%)	Malaysia	33 (0,3%)	Chile	1 (0,01%)
Serbia	383 (3,3%)	Morocco	116 (1,0%)	Nepal	51 (0,4%)	Mexico	29 (0,3%)	Sri Lanka	1 (0,01%)
Philippines	284 (2,5%)	United States	99 (0,9%)	Poland	51 (0,4%)	France	21 (0,2%)		

Table 60: Overview of the origins and distribution of all participating contributors.

To complete all tests for all of the three phases, 11534 contributions were received from countries all over the world as shown in Figure 60. The largest group of contributors come from Venezuela, which contributed just over 40% of all contributions. As discussed before, this can be attributed to the fact that Venezuelans are currently the largest group of people active on crowdsourcing platforms due to their dire economic situation (Posch et al., 2018). However, multiple larger groups of contributors were also observed coming from countries such as Turkey, Egypt, Russia, Ukraine, India and Indonesia. We have observed small indications that the time that the job launches has on the type of gathered crowd workers. However, no conclusive evidence could be found for the potential effect that the time of the day on the different groups of contributors and the overall quality of the results.

All of the provided contributions came from 603 unique contributors that were automatically accumulated for the total price of $354.72. For these rewards, they were able to complete all the work for all tests for all three phases in a total time of just 5,4 hours. This time does not take into account the time to set up the jobs of the platform and the time to construct the method itself. Nevertheless, the time required to complete the work by the crowd is estimated to be a tenfold quicker than doing it manually, comparing it with the required effort for the classification that enabled the comparison of the results by a single researcher. Furthermore, the method itself and the constructed job descriptions could be reused indefinitely for all different kinds of sets of online user reviews, rendering the time for the (further) construction of the method negligible.

Times Participated	Once	Twice	Thrice	Four Times	Five Times
Frequency	499	75	23	5	1

Table 61: Frequency of participating contributors between all five sessions of the three tests.

In regards of contributor retention, reoccurring contributor identifiers have been observed between all five test sessions. As shown in Table 61, 499 out of the total 603 unique contributors only contributed to one of the five sessions. Furthermore, we identified 75 contributors that contributed to two sessions and 23 contributors to three test sessions. More amazingly, instances were observed that indicate that 5 contributors contributed four times and even 1 out of the 603 contributors managed to contribute to all five of the test sessions. None of the test sessions were planned and were not announced beforehand, showing that at least a part of the contributors are very active on platforms for crowdsourcing purposes. These figures show that it may be possible to retain some of the crowd workers for future jobs, should they be posted somewhat regularly and are made recognizable. Worker retention may be beneficial as they gain more experience with the tasks and therefore will require less training and produce higher quality results in the future.

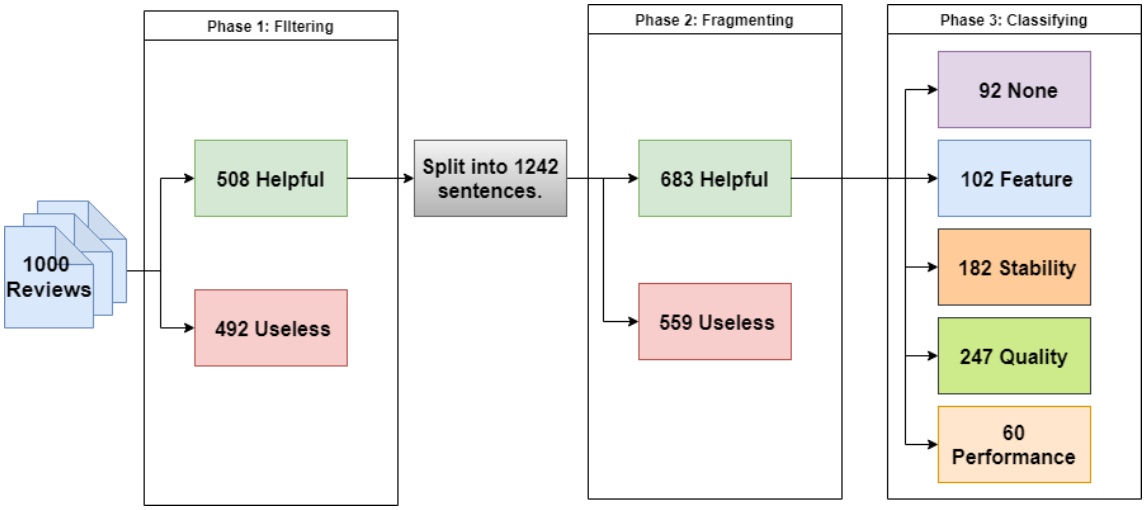


Figure 32: Overview of the course of the user reviews through the different phases.

In respect of the journey of the reviews throughout the entire method, Figure 32 shows what happened with the reviews in each of the executed phases. We initially started with 1,000 unprocessed reviews from 8 different products, straight from the Apple App store and the Google Play store. In phase 1, these were classified as 508 helpful and 492 useless reviews by the crowd. When comparing the performance of the crowd with the classification of the researcher, we discovered that the crowd was able to identify 84% of all useless results in the entire dataset. The 492 useless reviews were discarded from the dataset and were not used in successive phases. Unfortunately, 7% of the reviews that were discarded actually held some valuable information according to the researcher.

In preparation of the second phase, the 508 helpful reviews were split per sentence by using an online tool made available by the Northwestern University of Information Technology. [1] This resulted in 1242 individual review fragments that were ready to be used as input for the second phase.

Phase 2 used a similar setup as the first phase, but with a modified task description catered to single review sentences. During the execution of the tests for phase 2, the 1242 individual fragments were classified as 683 helpful fragments and 559 useless ones by the crowd. Again, the set of useless reviews were discarded as they had no further use in successive phases. When comparing the performance of the crowd with the golden standard created by the researcher, we discovered that 12% of the discarded sentences contained helpful. Furthermore, we discovered that the crowd was able to identify 81% of the useless reviews that the researcher identified. This meant that 19% of the fragments kept in the dataset were actually perceived as useless and continued as input for

[1] http://morphadorner.northwestern.edu/morphadorner/sentencesplitter

the third phase as well.

The third phase used the output of the second phase and aimed to classify those 683 helpful review fragments into five categories. This meant that the review fragments could be classified as either being a feature request, being stability, performance or quality related or none at all. The none category was defined to either catch useless fragments that remained in the dataset after the second phase or to provide an alternative when they did not match the other four categories.

The crowd classified those 683 fragments as 102 feature requests, 182 stability related, 60 performance related, 247 quality related and 92 into the none category. Compared with the golden standard, the crowd matched for 78% with the classification of the researcher. The crowd reached the highest precision and recall values for the feature, stability and quality categories. Points for improvement were identified for the performance and and none category, such as better distinctions between the performance and quality categories in the tasks description.

To make the final output of the method more tangible, examples of correctly classified review fragments are shown in Table 62.

None	**Feature**	**Stability**	**Performance**	**Quality**
My wife and I share business and personal notebooks across our iPhones, iPads and MacBooks.	I only wish there were options for font and background themes and such.	Freezes a lot and only allows so many followings per hour!	The most recent update improved the speed.	Very clean interface and easily accessible links.
I use this app seriously all the time.	Notifications are at the top of my wishlist, though.	I open it up and it closes on me.	This app is using a lot of battery.	Texts and sending pics worked great!
Not even worth the download time.	Color coding the tweets per user would be nice as well.	Please update so when I send a tweet it doesn't crash.	This app runs so slowly that it is essentially worthless.	Words overlap on my phone.

Table 62: Instances of correctly classified review fragments produced by the method.

For each of the five categories, three review fragments were chosen that represent the core aspect and purpose of its respective category. When linked to the product that these review fragments originate from, it is evident that helpful information was extracted from a large pile reviews that contained a lot of irrelevant information. Furthermore, this valuable information was extracted in a fraction of the time that the researcher needed to create their own classification of the same dataset for results comparison. Lastly, while the scope of this project limited the further refinement of these fragments and their categories, possibilities absolutely exist to extent the created method to refine the outcome even further.

7 Conclusion

The purpose of this thesis is characterized by three separate ambitions, encapsulated in the main research question and its subquestions. While the main research question mainly refers to the development of a crowdsourcing technique, elements of practical purpose and feasibility are also incorporated to prove its worth beyond the conceptual realm.

Referring to the purpose of the technique and its place in requirements engineering practices, the literature study aggregated evidence that supports the added value of a crowdsourcing approach over existing techniques. By investigating potential automated techniques from the natural language processing discipline, it could be concluded that existing automated means are not accurate enough in dealing with the challenges related to user generated feedback such as online user reviews. Crowdsourcing has been perceived by many as a potentially cheaper, faster and better scaling concept that keeps the human factor involved in the process, making it a suitable alternative to existing approaches. Subsequently, it appeared to be able to fulfill the desire from the requirements engineering discipline to involve feedback from a large and diverse group of users in a timely fashion. However, evidence was lacking that demonstrated that these benefits could actually be yielded in practice for tasks in the context of requirements engineering.

To maximize the chance to benefit from a crowdsourcing approach optimally, the main research question was investigated. Included in the second part of the literature review, all possible design approaches and required components were investigated to guide the design process. It resulted in a streamlined guidance framework that included concepts as crowd involvement, quality control, crowd training, incentives and task structures that demonstrates how a crowdsourcing technique could be developed. From these concepts, a first iteration emerged catered to both deal with the challenging nature of user reviews and the complex task of classifying requirements in multiple steps. Internal tests were conducted to further refine the method and to make it more resilient to context specific challenges. Results from the internal test were combined and processed in a move towards a final and well substantiated conceptual model.

Utilizing the final conceptual model, three tests were conducted to investigate the practical feasibility of a crowdsourcing approach for the purpose of requirements elicitation. These tests provided ample evidence that support the notion that this crowdsourcing solution is a feasible approach that has the capability to deal with the involved challenges when dealing with user generated feedback. The first test assessed whether crowd workers were able to correctly classify entire user reviews as helpful or useless compared to the created golden standard. It found that crowd workers marked reviews as useless with a precision score of 93%, meaning that only 7% of the reviews marked as useless would be wrongly discarded. Furthermore, the crowd was able to recall 84% of the useless reviews from the golden standard. The second test had similar objectives, only this time using the helpful reviews from the first test that were now split into individual sentences. The results showed that crowd workers were able to complete this job with an 88% precision rate, while identifying a total of 81% of all useless review fragments from the golden standard. The combination of the results from the first and second tests allows us to address the hypothesis stating that crowd workers have the ability to recognize the differences between useful and useless results. With the high rates of precision and recall that were reached with only three judgments per review or fragment on separate occasions, we can conclude that crowd workers indeed possess the required ability for this kind of job.

The third test is unique from the other two in the sense that it was more ambitious regarding expectations of capabilities from a group of crowd workers. Instead of classifying reviews or fragments thereof into two separate classes, crowd workers were now required to classify review fragments into five different requirements categories. Nevertheless, crowd workers were able to classify the review fragments with an accuracy of 74% compared to the golden standard with the feature and stability categories being the best performing ones. Furthermore, we were able to reach a perfect agreement of 100% with the classification of the golden standard when all six judgments were classified unanimously as the same category. These values diminished to 91%, 85%, 65% and 49% when one less contributor agreed with the majority each time respectively. Based on these results, the second hypothesis stating that crowd workers are able to correctly categorize user re-

views in different requirement categories can therefore also be accepted, although differences were observed between the different categories.

Now only the third and final hypothesis remains, stating that a crowdsourcing technique for the elicitation of requirements from online reviews is a feasible and cost-efficient approach. Considering the accuracy of the results from the tests combined with the total cost of $350 to fully process a set of 1,000 user reviews, we are inclined to conclude that the constructed approach is feasible and cost-effective. While the design of the method took up most of the time, the execution time is negligible as the entire method took only 5,4 hours to fully process 1,000 reviews over all three phases. In contrast, the creation of the golden standard for all three phases is estimated to have cost between twenty and thirty man hours. Most importantly however, the high reusability factor of this method highlights its value in the best way, as none of the phases are catered to specific (types of) applications. The entirety of the method and all of its individual components were constructed with the aim to handle user reviews from a variety of sources, and has shown to be perfectly capable of doing exactly this.

8 Discussion

Even with the carefully defined and executed approach constructed beforehand, not everything could be accounted for due to the experimental nature of this research. The design of the method had to rely on many assumptions in early phases due to lacking supportive literature that could guide the decision making process. As a consequence, we have no way of knowing whether our method reached its highest or lowest potential which makes it harder to place the results in context. Furthermore, it is currently impossible to trace back potential flaws of the method to individual design decisions, which impedes the refinement and improvement of the method. The potential improvements for the method that we could both identify and substantiate so far refer to the effectiveness of the training method.

Moreover, the tests were conducted while having access to only anecdotal experience with the Figure Eight platform. As a result, no restrictions were placed on specific countries or channels that could potentially be notorious for providing results with a significantly lower quality. The analysis of the outcome currently does not account or compensate for the possibility that these sources influenced the results. Not having traced back common errors or mistakes to their sources currently makes it impossible to prevent this from occurring in the future.

Regarding the acquired results, the executed phases utilized inputs from preceding phases, which means that errors of the crowd in earlier phases could have echoed throughout all tests. With this configuration we examined the performance of the whole method in its entirety, and not the performance of individual phases. Using the phases in a standalone fashion could therefore lead to deviations from the results observed in this research. Similarly, only three judgments were required for the classification of a review in the first two tests, and only six judgments were required for the third test. This left the majority of classifications up to only a very small part of the crowd. Combined with the notion that 100% agreement rating was reached when all six contributors agreed on the same classification during the third test, we have to assume that the quality of the results can be improved by involving more crowd workers. Although at this point in time, no precise conclusion can be drawn on this aspect. Finally, the dataset used for the tests contains reviews that were written between the year 2011 and 2015. This allows for questions to be raised regarding the representativeness of the reviews for current online environments.

Concluding, the creation of the golden standard and the execution of the outcome review in each analysis relied on the expertise and vision of one single researcher. While transparency has been offered into this process during the analysis, the initial classification of the researcher has not been cross-checked in its entirety. As the workload for the creation of all golden standards for all three tests was relatively high, it is not unreasonable to assume that some errors made it into the golden standards and therefore affected the validity of the results.

9 Future Work

The results from this thesis can function as the foundation for two different types of future work. First and foremost, the focus of the future work can be laid upon further evolving the constructed method. The performance of the constructed method could be investigated more in depth in order to further evolve the practice of eliciting requirements from online reviews through means of crowdsourcing. Multiple points were identified where the method could be potentially improved, being it an improved training method or the application of more strict quality controls. Further research could indicate whether the fixes for potential flaws affect the quality of the results either positively or negatively. Furthermore, the method has shown to be quite flexible during the design of the process. The method has the potential to handle multiple different variations deviating from our configuration. Possibilities exist where additional phases are added, changed or removed based on the intentions of a practitioner. It could be interesting to research the performance of more extensive configurations that results in even finer-grained results.

The constructed method could also be subjected to additional tests to see how the results hold up when a more strict golden standard is used for comparison, or when reviews from different apps, other sources or even entire datasets are used. Another underexposed characteristic of the method is its ability to scale with larger datasets, which will require additional research to fully investigate. Different analysis methods could also be deployed that for instance not take the aggregated answer of the crowd, but will take the answer given by the contributor with the highest level of skill or trust. Such selective aggregation of the results could result into new insights, while reusing the same data gathered during the tests of this research. Furthermore, more extensive analysis could be applied on the generated results from this project to better explore the performance of the method regarding reviews from individual products. Additionally, further research into the performance of the different types of contributors could also be conducted, either focused on their originating channels or geographical information. This will provide more in-depth knowledge on how to improve the effectiveness of the available quality control mechanisms.

Finally, the second area for future work is based on the notion that this research can be an advocate for the use of crowdsourcing solutions in either the field of requirements engineering or other disciplines. The results from this research are promising enough to warrant the exploration of crowdsourcing solutions for similar challenges that revolve around large volumes of data with a difficult nature. This research has shown that crowd workers are able to deal with perhaps more complex problems than initially assumed when they are properly instructed, making it possible to apply crowdsourcing solutions for more types of tasks than maybe originally assumed.

Acknowledgements

First and foremost I would like to thank both my supervisors for their support during this thesis project. I would like to thank dr. Fabiano Dalpiaz for his excellent coaching and constructive feedback, his personal interest in the project and his trust to allocate funds to test the experimental method that we constructed. I would like to thank dr. Ioanna Lykourentzou for her provided insight into the world of crowdsourcing and her experience with the Figure Eight platform.

Furthermore, I would like to express my gratitude to Eduard Groen from the Fraunhofer Institute for his involvement in this project. Both for his commitment to provide the dataset that was used to test the method, and for the fact that he was always readily available for discussions or extensive feedback.

Additionally I would like to thank Sabine Molenaar and the other people from the Requirements Engineering Lab at Utrecht University for their insightful discussions, available expertise and their assistance with LaTeX.

Concludingly, I would like to thank my parents and my girlfriend for their endless support and their sometimes much needed motivational speeches and supportive messages.

References

Adepetu, A., Ahmed, K. A., Al Abd, Y., Al Zaabi, A., & Svetinovic, D. (2012). Crowdrequire: A requirements engineering crowdsourcing platform. In *Aaai spring symposium: Wisdom of the crowd* (pp. 2–7).

Ali, R., Solis, C., Salehie, M., Omoronyia, I., Nuseibeh, B., & Maalej, W. (2011). Social sensing: when users become monitors. In *Proceedings of the 19th acm sigsoft symposium and the 13th european conference on foundations of software engineering* (pp. 476–479).

Allahbakhsh, M., Benatallah, B., Ignjatovic, A., Motahari-Nezhad, H. R., Bertino, E., & Dustdar, S. (2013). Quality control in crowdsourcing systems: Issues and directions. *IEEE Internet Computing*, *17*(2), 76–81.

Berry, D. (2001). Natural language and requirements engineering-nu. In *International workshop on requirements engineering, imperial college, london, uk.*

Berry, D., Gacitua, R., Sawyer, P., & Tjong, S. F. (2012). The case for dumb requirements engineering tools. In *International working conference on requirements engineering: Foundation for software quality* (pp. 211–217).

Berry, D., et al. (2000). From contract drafting to software specification: Linguistic sources of ambiguity-a handbook version 1.0.

Bigham, J. P., Bernstein, M. S., & Adar, E. (2015). Human-computer interaction and collective intelligence. *Handbook of collective intelligence*, *57*.

Bogers, M., Afuah, A., & Bastian, B. (2010). Users as innovators: A review, critique, and future research directions. *Journal of management*, *36*(4), 857–875.

Borromeo, R. M., & Toyama, M. (2016). An investigation of unpaid crowdsourcing. *Human-centric Computing and Information Sciences*, *6*(1), 11.

Bourque, P., Fairley, R. E., et al. (2014). *Guide to the software engineering body of knowledge (swebok (r)): Version 3.0.* IEEE Computer Society Press.

Charette, R. N. (2005). Why software fails [software failure]. *Ieee Spectrum*, *42*(9), 42–49.

Chemuturi, M. (2012). *Requirements engineering and management for software development projects.* Springer Science & Business Media.

Chen, J. J., Menezes, N. J., Bradley, A. D., & North, T. (2011). Opportunities for crowdsourcing research on amazon mechanical turk. *Interfaces*, *5*(3), 1.

Chen, N., Lin, J., Hoi, S. C., Xiao, X., & Zhang, B. (2014). Ar-miner: mining informative reviews for developers from mobile app marketplace. In *Proceedings of the 36th international conference on software engineering* (pp. 767–778).

Chung, L., & do Prado Leite, J. C. S. (2009). On non-functional requirements in software engineering. In *Conceptual modeling: Foundations and applications* (pp. 363–379). Springer.

Dalpiaz, F., & Parente, M. (2019). RE-SWOT: From User Feedback to Requirements via Competitor Analysis. In *Proceedings of the 25th international working conference on requirements engineering: Foundation for software quality (refsq'19).*

Dalpiaz, F., van der Schalk, I., & Lucassen, G. (2018). Pinpointing ambiguity and incompleteness in requirements engineering via information visualization and nlp. *In: International Working Conference on Requirements Engineering: Foundation for Software Quality*, 119–135.

Dalpiaz, F., van der Schalk, I., Brinkkemper, S., Aydemir, F. B., & Lucassen, G. (2018). Detecting terminological ambiguity in user stories: Tool and experimentation. *Information and Software Technology*.

Dalpiaz, F., Van Der Schalk, I., & Lucassen, G. (2018). Pinpointing ambiguity and incompleteness in requirements engineering via information visualization and nlp. In *International working conference on requirements engineering: Foundation for software quality* (pp. 119–135).

Dhinakaran, V. T., Pulle, R., Ajmeri, N., & Murukannaiah, P. K. (2018). App review analysis via active learning. In *International requirements engineering conference.*

Erickson, L., Petrick, I., & Trauth, E. (2012). Hanging with the right crowd: Matching crowdsourcing need to crowd characteristics.

Estellés-Arolas, E., Navarro-Giner, R., & González-Ladrón-de Guevara, F. (2015). Crowdsourcing fundamentals: definition and typology. In *Advances in crowdsourcing* (pp. 33–48). Springer.

Fernández, D. M., & Wagner, S. (2015). Naming the pain in requirements engineering: A design for a global family of surveys and first results from germany. *Information and Software Technology*, *57*, 616–643.

Finin, T., Murnane, W., Karandikar, A., Keller, N., Martineau, J., & Dredze, M. (2010). Annotating named entities in twitter data with crowdsourcing. In *Proceedings of the naacl hlt 2010 workshop on creating speech and language data with amazon's mechanical turk* (pp. 80–88).

Finkelstein, A., & Lim, S. L. (2012). Stakerare: using social networks and collaborative filtering for large-scale requirements elicitation. *IEEE transactions on software engineering*(3), 707–735.

Gao, G. G., Greenwood, B. N., Agarwal, R., & McCullough, J. S. (2015). Vocal minority and silent majority: how do online ratings reflect population perceptions of quality?

Geiger, D., Seedorf, S., Schulze, T., Nickerson, R. C., & Schader, M. (2011). Managing the crowd: Towards a taxonomy of crowdsourcing processes. In *Amcis.*

Genc-Nayebi, N., & Abran, A. (2017). A systematic literature review: Opinion mining studies from mobile app store user reviews. *Journal of Systems and Software*, *125*, 207–219.

Glinz, M. (2007). On non-functional requirements. In *Requirements engineering conference, 2007. re'07. 15th ieee international* (pp. 21–26).

Groen, E. C., Kopczyńska, S., Hauer, M. P., Krafft, T. D., & Doerr, J. (2017). Users—the hidden software product quality experts?: A study on how app users report quality aspects in online reviews. In *Requirements engineering conference (re), 2017 ieee 25th international* (pp. 80–89).

Groen, E. C., Schowalter, J., Kopczynska, S., Polst, S., & Alvani, S. (2018). Is there really a need for using nlp to elicit requirements? a benchmarking study to assess scalability of manual analysis.

Groen, E. C., Seyff, N., Ali, R., Dalpiaz, F., Doerr, J., Guzman, E., ... others (2017). The crowd in requirements engineering: The landscape and challenges. *IEEE software*, *34*(2), 44–52.

Hazzan, O., Lapidot, T., & Ragonis, N. (2015). *Guide to teaching computer science: An activity-based approach.* Springer.

Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. *Science*, *349*(6245), 261–266.

Horton, J. J., & Chilton, L. B. (2010). The labor economics of paid crowdsourcing. In *Proceedings of the 11th acm conference on electronic commerce* (pp. 209–218).

Hossain, M. (2012). Crowdsourcing: Activities, incentives and users' motivations to participate. In *2012 international conference on innovation management and technology research* (pp. 501–506).

Hosseini, M., Groen, E. C., Shahri, A., & Ali, R. (2017). Craft: A crowd-annotated feedback technique. In *2017 ieee 25th international requirements engineering conference workshops (rew)* (pp. 170–175).

Hosseini, M., Phalp, K. T., Taylor, J., & Ali, R. (2014). Towards crowdsourcing for requirements engineering.

Howe, J. (2006). The rise of crowdsourcing. *Wired magazine*, *14*(6), 1–4.

Inayat, I., Salim, S. S., Marczak, S., Daneva, M., & Shamshirband, S. (2015). A systematic literature review on agile requirements engineering practices and challenges. *Computers in human behavior*, *51*, 915–929.

Jeppesen, L. B., & Lakhani, K. R. (2010). Marginality and problem-solving effectiveness in broadcast search. *Organization science*, *21*(5), 1016–1033.

Johnson, R. T., & Johnson, D. W. (2008). Active learning: Cooperation in the classroom. *The annual report of educational psychology in Japan*, *47*, 29–30.

Kassab, M., Neill, C., & Laplante, P. (2014). State of practice in requirements engineering: contemporary data. *Innovations in Systems and Software Engineering*, *10*(4), 235–241.

Kittur, A., Smus, B., Khamkar, S., & Kraut, R. E. (2011). Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th annual acm symposium on user interface software and technology* (pp. 43–52).

Kof, L. (2004). *Natural language processing for requirements engineering: Applicability to large requirements documents.* Citeseer.

LaToza, T. D., Towne, W. B., Van Der Hoek, A., & Herbsleb, J. D. (2013). Crowd development. In *Cooperative and human aspects of software engineering (chase), 2013 6th international workshop on* (pp. 85–88).

LaToza, T. D., & van der Hoek, A. (2016). Crowdsourcing in software engineering: Models, motivations, and challenges. *IEEE software*, *33*(1), 74–80.

Lévy, P., & Bononno, R. (1997). *Collective intelligence: Mankind's emerging world in cyberspace.* Perseus books.

Liddy, E. D. (2001). Natural language processing.

Lu, M., & Liang, P. (2017). Automatic classification of non-functional requirements from augmented app user reviews. In *Proceedings of the 21st international conference on evaluation and assessment in software engineering* (pp. 344–353).

Maalej, W., Nayebi, M., Johann, T., & Ruhe, G. (2016). Toward data-driven requirements engineering. *IEEE Software*, *33*(1), 48–54.

Malone, T. W., Malone, T. W., & Crowston, K. (1994). The interdisciplinary study of coordination. *ACM Computing Surveys (CSUR)*, *26*(1), 87–119.

Manning, C. D., Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT press.

Mao, K., Capra, L., Harman, M., & Jia, Y. (2015). A survey of the use of crowdsourcing in software engineering. *Rn*, *15*(01).

Mason, W., & Watts, D. J. (2010, May). Financial incentives and the "performance of crowds". *SIGKDD Explor. Newsl.*, *11*(2), 100–108. Retrieved from http://doi.acm.org/10.1145/1809400.1809422 doi: 10.1145/1809400.1809422

Mumford, M. D. (2003). Where have we been, where are we going? taking stock in creativity research. *Creativity research journal*, *15*(2-3), 107–120.

Nuseibeh, B., & Easterbrook, S. (2000). Requirements engineering: a roadmap. In *Proceedings of the conference on the future of software engineering* (pp. 35–46).

och Dag, J. N., Gervasi, V., Brinkkemper, S., et al. (2005). A linguistic-engineering approach to large-scale requirements management. *IEEE software*(1), 32–39.

och Dag, J. N., Gervasi, V., Brinkkemper, S., & Regnell, B. (2004). Speeding up requirements management in a product software company: Linking customer wishes to product requirements through linguistic engineering. In *Requirements engineering conference, 2004. proceedings. 12th ieee international* (pp. 283–294).

Paetsch, F., Eberlein, A., & Maurer, F. (2003). Requirements engineering and agile software development. In *Enabling technologies: Infrastructure for collaborative enterprises, 2003. wet ice 2003. proceedings. twelfth ieee international workshops on* (pp. 308–313).

Pagano, D., & Maalej, W. (2013). User feedback in the appstore: An empirical study. In *2013 21st ieee international requirements engineering conference (re)* (pp. 125–134).

Pang, B., Lee, L., et al. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, *2*(1–2), 1–135.

Paolacci, G., Chandler, J., & Ipeirotis, P. G. (2010). Running experiments on amazon mechanical turk. *Judgment and Decision making*, *5*(5), 411–419.

Parvanta, C., Roth, Y., & Keller, H. (2013). Crowdsourcing 101: a few basics to make you the leader of the pack. *Health promotion practice*, *14*(2), 163–167.

Penalver-Martinez, I., Garcia-Sanchez, F., Valencia-Garcia, R., Rodriguez-Garcia, M. A., Moreno, V., Fraga, A., & Sanchez-Cervantes, J. L. (2014). Feature-based opinion mining through ontologies. *Expert Systems with Applications*, *41*(13), 5995–6008.

Pohl, K. (1994). The three dimensions of requirements engineering: a framework and its applications. *Information systems*, *19*(3), 243–258.

Posch, L., Bleier, A., Flöck, F., & Strohmaier, M. (2018). Characterizing the global crowd workforce: A cross-country comparison of crowdworker demographics. *arXiv preprint arXiv:1812.05948*.

Preece, J., Rogers, Y., & Sharp, H. (2015). *Interaction design: beyond human-computer interaction*. John Wiley & Sons.

Qi, J., Zhang, Z., Jeon, S., & Zhou, Y. (2016). Mining customer requirements from online reviews: A product improvement perspective. *Information & Management*, *53*(8), 951–963.

Quinn, A. J., & Bederson, B. B. (2011). Human computation: a survey and taxonomy of a growing field. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 1403–1412).

Ramesh, B., Cao, L., & Baskerville, R. (2010). Agile requirements engineering practices and challenges: an empirical study. *Information Systems Journal*, *20*(5), 449–480.

Retelny, D., Robaszkiewicz, S., To, A., Lasecki, W. S., Patel, J., Rahmati, N., ... Bernstein, M. S. (2014). Expert crowdsourcing with flash teams. In *Proceedings of the 27th annual acm symposium on user interface software and technology* (pp. 75–85).

Ryan, K. (1993). The role of natural language in requirements engineering. In *Proceedings of the ieee international symposium on requirements engineering* (pp. 240–242).

Schenk, E., & Guittard, C. (2011). Towards a characterization of crowdsourcing practices. *Journal of Innovation Economics & Management*(1), 93–107.

Silberman, M. (1996). *Active learning: 101 strategies to teach any subject.* ERIC.

Smith, A. (2014). Requirements management: A core competency for project and program success..

Snijders, R., Dalpiaz, F., Hosseini, M., Shahri, A., & Ali, R. (2014). Crowd-centric requirements engineering. In *Utility and cloud computing (ucc), 2014 ieee/acm 7th international conference on* (pp. 614–615).

Valentine, M. A., Retelny, D., To, A., Rahmati, N., Doshi, T., & Bernstein, M. S. (2017). Flash organizations: Crowdsourcing complex work by structuring crowds as organizations. In *Proceedings of the 2017 chi conference on human factors in computing systems* (pp. 3523–3537).

Vukovic, M., & Bartolini, C. (2010). Towards a research agenda for enterprise crowdsourcing. In *International symposium on leveraging applications of formal methods, verification and validation* (pp. 425–434).

Wexler, M. N. (2011). Reconfiguring the sociology of the crowd: exploring crowdsourcing. *International Journal of Sociology and Social Policy*, *31*(1/2), 6–20.

Williams, G., & Mahmoud, A. (2017). Mining twitter feeds for software user requirements. In *2017 ieee 25th international requirements engineering conference (re)* (pp. 1–10).

Zave, P. (1997, December). Classification of research efforts in requirements engineering. *ACM Comput. Surv.*, *29*(4), 315–321. Retrieved from http://doi.acm.org/10.1145/267580.267581 doi: 10.1145/267580.267581

Zhang, H., Horvitz, E., Miller, R. C., & Parkes, D. C. (2011). Crowdsourcing general computation.

Zhang, Y., Witte, R., Rilling, J., & Haarslev, V. (2006). An ontology-based approach for traceability recovery. In *3rd international workshop on metamodels, schemas, grammars, and ontologies for reverse engineering (atem 2006), genoa* (pp. 36–43).