Involvement of Multiple Stakeholders in Agile Software Development

Master Thesis

Menan Rajasegaram
Nicolai Vork
Søren Lønsman Larsen

Aalborg University
Department of Computer Science
Selma Lagerlöfs Vej 300
DK-9220 Aalborg Ø

SICT
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Involvement of Multiple Stakeholders in Agile Software Development

Theme:
Systems Development

Project Period:
Spring Semester 2018

Project Group:
hcc107f18

Participant(s):
Menan Rajasegaram
Nicolai Vork
Søren Lønsman Larsen

Supervisor(s):
Jan Stage

Page Numbers: 54

Date of Project Submission:
June 20, 2018

Abstract:

This project concerns the involvement of multiple stakeholders in software development. This report consists of three academic papers. The first paper presents an exploratory literature review on existing research concerning involvement of multiple stakeholders in agile software development. In total, 15 paper were identified and used in the review. Based on the review, we found that there should exist a supporting structure for involvement of stakeholders. Further, stakeholders should be involved early in development and continuously throughout development. The second paper concerns a survey on how the software industry involves multiple stakeholders during development, and the challenges that the industry experiences. In total, 50 responses were used in the results. We found that the most used method for involvement of stakeholders was meetings. Further, the most common challenge found was conflicting requirements. The third paper presents an empirical study on the involvement of multiple stakeholders in a development process based on Scrum. We involved several Product Owners in both Sprint Planning and Sprint Reviews and involved stakeholders in Sprint Reviews. Some of the challenges experienced were lack of time for meetings and lack of feedback on usability.

Summary

Siden det agile manifest blev publiceret i 2001, er der opstået mange agile udviklingsmetoder, heriblandt Scrum, Kanban og Extreme Programming. Fælles for disse udviklingsmetoder er, at de bygger på den grundlæggende værdi i det agile manifest, at man værdsætter samarbejde med kunden mere end at forhandle om en kontrakt.

Udover kunden findes der imidlertid også andre interessenter i et udviklingsprojekt, som kan være fordelagtige at involvere og samarbejde med. Eksempler på disse interessenter er slutbrugere, sponsorer, domæne eksperter samt partnere.

Et udviklingsprojekt hvori flere interessenter er involveret kan dog give anledning til en række problemer. Forskellige interessenter kan være i konflikt med hinanden, og det kan derfor være svært eller enddog umuligt at tilgodese alle interessenter. Dertil har interessenter ofte svært ved at formulere sine ønsker og behov, hvilket kan gøre det yderligere besværligt at involvere og samarbejde med dem.

Baseret på dette formulerede vi følgende forskningsspørgsmål "Hvordan kan flere interessenter involveres i agil udvikling?"

For at besvare spørgsmålet foretog vi tre studier: Et eksplorativt litteraturstudie, en spørgeundersøgelse samt et case studie.

Det eksplorative litteraturstudie havde til formål at besvare delspørgsmålet "Hvordan foreslår litteraturen at flere interessenter bør involveres under agil udvikling?"

I litteraturstudiet inspicerede vi 15 artikler, som vi inddelte i kategorierne HCD i Scrum, kommunikation i agil udvikling, kravudledning, og projekt beslutningstagning. Vi konkluderede at der bør være en struktur der understøtter involvering af flere interessenter, at interessenter bør involveres tidligt og gennem hele udviklingsprocessen og at konflikter mellem interessenter bør håndteres.

Spørgesundersøgelsen havde til formål at besvare delspørgsmålet "Hvordan involverer software industrien flere interessenter under udvikling, og hvilke udfordringer oplever industrien?"

I spørgeundersøgelsen fik vi 50 brugbare besvarelser. Vi spurgte ind til hvilke metoder der blev brugt til involvering af interessenter, og hvilke udfordringer der

var ved involvering af flere interessenter. Vi konkluderede, at industrien primært bruger møder til at involvere interessenter, at industrien oplever udfordringer ved involvering af flere interessenter, og at industrien imødekommer udfordringerne på forskellig vis.

Slutteligt havde casestudiet til formål at besvare delspørgsmål "Hvordan er det muligt at involvere flere interessenter i en udviklingsmetode baseret på Scrum, og hvilke udfordringer er der ved det?"

I casestudiet involverede vi flere interessenter i udvikling af en app til patienter. Disse interessenter var patienter, læger, farmaceuter og lægesekretærer.

Vi konkluderede, at det er muligt at involvere flere interessenter i en udviklingsmetode baseret på Scrum. Vi fandt også en del udfordringer ved involvering af flere interessenter såsom begrænset tid til afholdelse af møder, begrænset feedback på usability, og konflikt mellem sprint mål og prioriteringer.

Baseret på disse tre studier konkluderede vi på forskningsspørgsmålet, at der er behov for strukturer, som understøtter involvering af flere interessenter, at involvering af flere interessenter er muligt gennem afholdelse af møder med interessenterne samt at der er udfordringer ved involvering af flere interessenter i agile metoder.

Contents

Preface

This report is written by three software engineering students as their Master Thesis project at Aalborg University.

The report consists of 4 chapters: Chapter 1 introduces the subject under research and the research questions. Further, a description of our case is presented. Chapter 2 sums up the contributions of three academic papers written by us. The three academic papers are placed in the appendix. Chapter 3 discusses the research methods that we used in the three academic papers. Finally, Chapter 4 concludes on the research questions, and presents limitations of our work and future work.

We would like to thank our supervisor Jan Stage for his supervision and guidance throughout the project.

Aalborg University, June 20, 2018

Chapter 1

Introduction

Agile Software Development (ASD) has in the last 25 years evolved with many new ways to develop software. Some of the early development methods that have emerged are Scrum in 1995 [9] and Extreme Programming (XP) in 1999 [1]. Common for the agile development methods is that they draw from some or all of the agile values and principles of the Agile Manifesto [2]. The general idea behind the agile manifesto is to be flexible and work with individuals over using plans and documentation.

One of the values of the manifesto is *customer collaboration* [2], where there is a focus on involving and collaborating with the customer throughout the development and not just before or after the development. Scrum attempts to achieve this through the Product Owner, who represents any stakeholders in the project including the customer and their needs [8]. XP goes even further and requires an on-site customer that is available to the development team throughout the development [1]. Although the customer seems to be involved heavily in agile development methods, there are other stakeholders, such as the end user, who do not receive the same attention as the customer with explicit structures or processes for involvement and collaboration with these.

This leads to the following problem statement:

> Problem Statement: How can multiple stakeholders be involved in agile software development?

In order to address the subject and answer the problem statement, we want to review existing literature on the subject. Further, we want to examine how the software industry involves multiple stakeholders. Finally, we want to perform an empirical study to create our own experiences with the subject. Consequently, we formulate 3 research questions. The first is as follows:

This paper presents an exploratory literature review of how multiple stakeholders can be involved in agile software development.

> 1st Research Question: How does the literature suggest that multiple stakeholders are involved in agile software development?

This research question addresses how to involve multiple stakeholders in agile software development through an exploratory literature review. If possible, we want to use relevant knowledge gained from the literature review for our empirical study on the subject. The second research question is the following:

> 2nd Research Question: How does the software industry involve multiple stakeholders during development, and what challenges does it experience?

This research question examines how the software industry involves multiple stakeholders, and which challenges they experience through a survey. We also want to use any knowledge from the survey in our empirical study if possible. The final research question is:

> 3rd Research Question: How can we involve multiple stakeholders in a development process based on Scrum, and which challenges are experienced?

This research question addresses specifically how Scrum can handle involvement and collaboration with multiple stakeholders during development. To answer this research question, we perform an case study, where we use a development method based on Scrum to develop a system. The case is described in the following section.

1.1 Case

The case revolves around prescription of medication in the primary health care. Specifically, we are concerned with the interactions that happen between patients, their general practitioner, medical secretaries, and pharmacists. We refer to these as the stakeholders.

The four stakeholders communicate heavily with each other during a patient's course of treatment. When a patient realises they have some symptoms, they call their medical practice to schedule an appointment through the medical secretary. At the appointment, the patient explains the symptoms they are experiencing to

their general practitioner, and the general practitioner attempts to diagnose the patient and prescribe medication if possible. The patient then uses the prescription to pick up the prescribed medication from a pharmacy through a pharmacist. Although the patient seems to be the center of communication between the stakeholders, the other stakeholders interact closely with each other to ensure the best treatment for the patient. For instance, the pharmacist may do check-ups on a prescription with the patient's general practitioner, if the prescription seems inadequate.

In order to improve the course of treatment for patients, we focus on developing an smartphone app for patients. The goal of the app is to empower patients with information and actions they can take to positively affect their course of treatment. Furthermore, the app may also alleviate the workload of general practitioners, medical secretaries, and pharmacists, since patients can receive information or perform actions through the app that they would otherwise have to contact the other stakeholders about.

Chapter 2

Contributions

In this Chapter we present the contributions of the three academic papers. The papers are included in the Appendix A.

2.1 Contribution 1

Menan Rajasegaram, Nicolai Vork, and Søren Lønsman Larsen. Involvement of Multiple Stakeholders in Agile Software Development: An Exploratory Literature Review. *Department of Computer Science, Aalborg University*. Aalborg, 2018.

This contribution presents an exploratory literature review on how multiple stakeholders can be involved in agile software development.

The study consisted of an exploratory literature review. The review initially yielded 56 articles. Out of the 56 articles, 15 articles passed all the inclusion criteria and were thus reported in the article. We used an affinity diagram to categorise the articles into the categories HCD in Scrum, Communication in Agile Development, Requirements Elicitation, and Project Decision Making.

Based on the reviewed articles, we concluded that a supporting structure for involvement of multiple stakeholders is needed in agile software development. The literature specifically points to Scrum as lacking this. Furthermore, stakeholders should be involved early and throughout the development. Thus, the developers are able to continuously validate the product with stakeholders throughout development and potentially reduce the cost. Lastly, conflicts between stakeholders should be handled to ensure satisfaction of all stakeholders. To this the literature suggests resolving conflicts through communication means or mathematical models that aim to find solutions that achieve consensus among stakeholders.

2.2 Contribution 2

Menan Rajasegaram, Nicolai Vork, and Søren Lønsman Larsen. A Survey on Involvement of Multiple Stakeholders in the Software Industry. *Department of Computer Science, Aalborg University*. Aalborg, 2018.

This contribution presents a survey on how the software industry involves multiple stakeholders during development, and the challenges that the industry experiences.

We conducted five semi-structured interviews with five different companies to gain an initial understanding of how they involve multiple stakeholders. We then created an initial web questionnaire that we piloted with three other companies. After this, we revised the questions in the questionnaire based on the feedback received from the piloting of the questionnaire. Finally, we distributed the web questionnaire on several online forums related to software development and contacted 12 companies directly. In total, 530 questionnaires were distributed of which we received 61 responses. 31 of these were fully completed, and the remaining 30 were partially completed. We removed 10 responses, since they only completed the first page of the questionnaire about the respondents' demographics, and one other response, because it was spam. Consequently, 50 responses were analysed.

Based on the results from the survey, we found that the most common challenge when involving multiple stakeholders faced by 17 out of 30 respondents was *conflicting requirements among stakeholders* followed by *misunderstanding the stakeholders*, which was experienced by 16 out of 30 respondents. Furthermore, we found that these challenges are accommodated differently. For instance, one respondent used refinement meetings for handling the challenge *conflicting requirements among stakeholders*, and another respondent used workshops for handling the same challenge. We also found that *meetings* was the most used method for involving stakeholders during requirements analysis, design, implementation, testing, and maintenance.

2.3 Contribution 3

Menan Rajasegaram, Nicolai Vork, and Søren Lønsman Larsen. Involvement of Multiple Stakeholders in Scrum: A Case Study on Prescription of Medication. *Department of Computer Science, Aalborg University*. Aalborg, 2018.

This contribution presents an empirical study of the possible involvement of multiple stakeholders and challenges that are experienced when multiple stakeholders are involved in a development process based on Scrum.

We conducted a case study on the case presented in Section 1.1. The case study had two parts. In the first part we elicited requirements for the app, and in the

second part we designed, implemented, and evaluated it. Throughout the development of the app, we involved patients, general practitioners, medical secretaries, and pharmacists as both Product Owners and stakeholders. The Product Owners were involved in the Sprint Planning and Sprint Review meetings, whereas the stakeholders were involved in the Sprint Review meetings.

We found that it is possible to involve multiple stakeholders in a development process based on Scrum. This was achieved through the involvement of several Product Owners, who each represented a specific stakeholder, and involvement of stakeholders throughout the entire development.

The involvement of multiple stakeholders caused a number of challenges. Only four of these are described below, but all of them are described in the article:

- Lack of time for meetings: Since the stakeholders only wanted to participate for a limited time at meetings, it became difficult to go over everything that was relevant to discuss at the meetings. Consequently, fewer requirements were processed in each Sprint to allow enough time to discuss them.

- Lack of feedback on usability: We rarely received feedback on the usability of the app, since the majority of their feedback referred to whether the requirements satisfied their needs and not how easy to use the implementation of them were.

- Difficult to set Sprint Goals: The Product Owners often set long-termed goals and repeated them rather than short-termed Sprint Goals. Thus, the goals were not helpful in deciding if a Sprint was successful.

- Conflicting Sprint Goals and prioritisations: The Sprint Goals and prioritisations were sometimes conflicting. This made it difficult or impossible to select requirements that satisfy both the Sprint Goals and prioritisations. As a result, a Sprint Goal or a highly prioritised requirement was sometimes not satisfied.

Chapter 3

Research Methods

In this Chapter we discuss the research methods we have used in the articles. In each of the sections we first present the method followed by their strengths and weaknesses. We then describe how we used the methods and strengths and weaknesses of this.

3.1 Literature Review

According to [4], *"a literature review is an objective, thorough summary and critical analysis of the relevant available research and non-research literature on the topic being studied."* [4].

A literature review is used to gain insight in an area through critique and summary of existing literature on the area [4]. We explain three types of literature reviews: Traditional, systematic, and exploratory literature reviews.

A traditional review summarises the existing body of literature. This type of review is selective about which literature becomes a part of the review, such as having some criteria for inclusion of literature. It is used to gain background information about a body of study and can be used to find holes in the theory in order to define new research questions in the area [4].

A systematic literature review is a more meticulous review. The literature review should present criteria for the formulation of the research question, criteria or a criterion for inclusion or exclusion of literature, criteria for the selection and access of the literature, criteria for reviewing the quality of the literature, and criteria for analysing the findings. The purpose of this type of review is to examine the complete body of knowledge in an area of study [4].

An exploratory literature review is for exploring research areas such as those presented in [5] and [7]. The review does not seek to cover all literature in a research area, but seeks to provide an overview as in [10]. An exploratory literature review uses exclusion criteria to limit the number of papers reviewed [5] [10] [7].

Strengths	Weaknesses
Gains insight into a subject	Time consuming
Provides rich data	Quality depends on criteria
	Subjective evaluation of literature

Table 3.1: Strengths and weaknesses of literature reviews.

The strengths and weaknesses of a literature review are shown in Table 3.1. The strengths are that a review can be used to gain insight into an area and provide a rich understanding of the subject under research. The weaknesses are that a review is time consuming, especially a systematic review. Further, the quality of a review is dependent on the criteria, and how subjective the evaluation of the literature is.

In the first article listed in Appendix A, we conducted an exploratory literature review of how to involve multiple stakeholders in agile software development. We used criteria for inclusion on the stages in the conducting phase of the review and used forward and backward linking to find additional literature.

The strength was that the literature review provided knowledge of the research area for future research. The weakness was that we did not conduct a systematic review of the literature. Therefore some of the relevant literature may not have been found. However, through the use of forward and backward linking, we alleviated this weakness. Another weakness was that the literature could be prone to subjective evaluations. We alleviated this weakness by having all of us review the literature.

3.2 Survey

According to [3], *"a survey is any activity that collects information in an organised and methodical manner. It is usually motivated by the need to study the characteristics of a population, build a database for analytical purposes or test a hypothesis."* [3].

The different types of surveys and their strengths and weaknesses are shown in Table 3.2.

An online survey is preferable, when one wants to reach a wide population group, such that results can be abstracted and generalised on a large population. For participants it is also easy to participate, since most people have access to the internet. Since the survey is online, the answers are also unbiased, since there is no interviewer [3]. One can use an online tool for making the surveys, for instance SurveyXact, which can simplify the process of collecting and analysing the responses.

An online survey also has disadvantages. The survey is open to anyone, when it is accessible online. Thus, one cannot control who the recipients are. Further, an online survey is vulnerable to technical difficulties when using a survey tool or

	Strengths	Weaknesses
Paper-Based	- Printed reference material can reduce response error - Computers not required for data collection	- Printing questionnaires can be expensive - Data analysis is separate from collection - Questionnaires cannot have skips - Questionnaires cannot be changed dynamic - Requires manual work for data collection - Questionnaires can only have simple edits
Online survey	- Edits during collection is automated - Data collection and analysis can be combined - Questionnaires can have complex branching - Easy to create statistics on data collection - Potentially cheaper than paper surveys - Anonymous data collection is easier - Environmentally friendly	- Digital collection must be thoroughly tested - Requires computer experts to develop - Vulnerable to technical difficulties - Respondents must know how to complete the survey - Questionnaires cannot be changed dynamic

Table 3.2: Strengths and weaknesses of the different types of surveys.

other services [3].

In the second article listed in Appendix A we used an online survey to question IT professionals on how they involve multiple stakeholders, and which challenges they experience when involving multiple stakeholders.

A strength of our survey was that the responses were anonymous, which encouraged respondents to be honest. Further, the distribution and collection of responses was automated, and the survey could skip questions, which were not relevant to certain respondents depending on how they answered previous questions. Another strength is that we used SurveyXact to collect and analyse data, which sped up the this the data collection and analysis process. However, our survey was vulnerable to technical difficulties, since we used SurveyXact to collect and analyse responses. Another weakness was that it was not possible for us to ask follow-up questions. This weakness was alleviated by conducting semi-structured interviews with IT professionals before creating the questionnaire to identify relevant questions. We also piloted the questionnaire to ensure the questions yielded usable responses.

3.3 Case Study

[11] defines a case study's scope as *"an empirical method that investigates a contemporary phenomenon (the "case") in depth and within its real-world context, especially when the boundaries between phenomenon and context may not be clearly evident."* [11]. A case study can be exploratory, descriptive, or explanatory.

In an exploratory case study researchers seek to explore a phenomenon by actively engaging in the context of the phenomenon. In this method the researchers might actively employ their theories in order to test them on the case. Through the study they test a theory and refine the method, in order to validate their theory or method [11].

In a descriptive case study researchers seek to describe a phenomenon. This can be done by having the researcher inserting themselves into the context of the phenomenon. The researcher can then describe how the phenomenon occurs within the context and present the evidence [11].

In an explanatory case study the researchers try to understand a case, which has already occurred. Firstly, questions for the study are defined. Then the researchers evaluate materials related to the case in order to construct a theory, which can explain why something has occurred. Lastly, the researcher can present an explanation based on the evidence found, and how it relates to other cases [11].

The strengths and weaknesses of a case study are shown in Table 3.3. The strengths of a case study are that it generates rich data, and is set in a natural setting. The weaknesses are that case studies are time consuming, since it studies a phenomenon in detail [6]. Further, depending on the case it can be difficult to

Strengths	Weaknesses
Natural settings	Time consuming
Rich data	Limited generalisation

Table 3.3: Strengths and weaknesses of a case study.

generalise with a single case study. However, this weakness can be alleviated with more case studies.

In the third article listed in Appendix A we conducted an exploratory case study. The study examined the possible involvement of multiple stakeholders in a development process based on Scrum, and the challenges that are experienced when involving multiple stakeholders. We conducted meetings with stakeholders to evaluate and gather feedback on the app.

The strength of our case study was the rich data gained from the evaluations with the different stakeholders. The weakness was that it was very time consuming to conduct the meetings.

Chapter 4

Conclusion

In this Chapter we conclude on the contributions of the papers in Appendix A followed by limitations and future work.

4.1 Research Questions

1st Research Question: How does the literature suggest that multiple stakeholders are involved in agile software development?

Based on the results from the literature review, we found that there is a need of a supporting structure for involvement of multiple stakeholders in agile software development. For instance, the literature shows that there exists variations of Scrum that incorporate stakeholders and their perspectives. This indicates that there is a need for a structure in Scrum that supports involvement of multiple stakeholders. Further, we found that stakeholders should be involved early and throughout the development to potentially reduce the cost. Lastly, in order to ensure satisfaction of stakeholders, conflicts among these should be handled. The literature suggests that communication means or mathematical models are ways of to resolve conflicts between the stakeholders.

2nd Research Question: How does the software industry involve multiple stakeholders during development, and what challenges does it experience?

Based on the results from the survey we conclude that there are different methods for involving multiple stakeholders. The most common method to involve multiple stakeholders during requirements analysis, design, implementation, testing, and maintenance was meetings. Further, the respondents experienced different challenges during development. According to the results, respondents experienced conflicting requirements among stakeholders as the most common challenge when involving multiple stakeholders. This was experienced by 17 out of 30 respondents. Further, they experienced misunderstanding the stakeholders as the

second most common challenge, and lack of engagement from stakeholders as the third most common challenge. The challenges are handled differently by the respondents. For instance, in order to accommodate conflicting requirements among stakeholders one respondent used workshops, and another used refinement meetings.

3rd Research Question: How can we involve multiple stakeholders in a development process based on Scrum, and which challenges are experienced?

Based on the gained experiences from the case study, we conclude that it is possible to involve multiple stakeholders in a development process based on Scrum. We achieved this with several Product Owners, who each represent a particular stakeholder. The Product Owners were involved in Sprint Planning and Sprint Review meetings, where they prioritised requirements, expressed Sprint Goals, and provided feedback on the implemented requirements. In addition, stakeholders were involved in Sprint Review meetings, where they provided feedback on implemented requirements. We conveyed information between stakeholders and Product Owners, such that the Product Owners were informed about the feedback from stakeholders.

From these involvements, we experienced seven challenges. First, we experienced lack of time for meetings, since stakeholders and Product Owners were not able to participate as much as we would like. Second, the stakeholders rarely expressed usability concerns, and instead focused on providing feedback on whether the requirements satisfied their needs. Third, the Product Owners had trouble expressing short-termed goals for Sprints, and instead provided long-termed ones, which they tended to repeat. Fourth, there were conflicting Sprint Goals and prioritisations, which caused difficulty with selecting requirements for Sprints. Fifth, we experienced a delay in reaching agreements, since stakeholders and Product Owners could not communicate directly due to separated meetings. Sixth, certain stakeholders became exhausted for new ideas that could potentially become new requirements. Seventh, we experienced difficulty involving patients, since they have specific interests and cannot represent other patients. Thus, a patient could not be assigned as a Product Owner.

Problem Statement: How can multiple stakeholders be involved in agile software development?

To answer the problem statement, we conducted three studies on involvement of multiple stakeholders: An exploratory literature review, a survey, and a case study.

From the literature review we concluded that there is a need of structures that can support involvement of multiple stakeholders in agile development methods, where the literature specifically addresses this need in Scrum. This correlates with our case study, which required modifications to Scrum in order to facilitate in-

volvement of multiple stakeholders. Thus, we conclude that structures should be established to involve multiple stakeholders in agile software development.

Furthermore, we concluded based on the the survey that the most common method for involving stakeholders during development and maintenance was meetings. In our case study we also used meetings to involve stakeholders during our development of the app. Therefore, we conclude that it is possible to involve multiple stakeholders through meetings.

In the case study we experienced some of the challenges that were experienced by the respondents in our survey. These were conflicting requirements, which was the most common challenge experienced by the respondents, and lack of engagement. In our case study conflicting requirements was experienced when involving several Product Owners to prioritise the requirements. Further, lack of engagement was experienced when involving stakeholders in Sprint Review meetings. Thus, we conclude that there are challenges when involving multiple stakeholders in agile software development.

4.2 Limitations

A limitation of our work is that the studies were conducted by us and also evaluated by us. Consequently, there may have been a bias towards presenting and highlighting certain results.

4.3 Future Work

Future work should look into how to alleviate the challenges of involving multiple stakeholders found in the studies. Further, other research methods could be employed to confirm the results from our studies.

Bibliography

[1] Kent Beck and Erich Gamma. *Extreme programming explained: embrace change*. addison-wesley professional, 2000.

[2] Kent Beck et al. "Manifesto for agile software development". In: (2001).

[3] Statistics Canada. *Survey methods and practices*. Ottawa - Ontario: Statistics Canada, 2003.

[4] P Cronin, F Ryan, and M Coughlan. "Undertaking a literature review: a step-by-step approach." In: *British Journal of Nursing* 17.1 (2008), pp. 38 –43.

[5] Uzair Iqbal Janjua, Alan Oxley, and Jafreezal Bin Jaffer. "Effective Risk Management of Software Projects (ERM): An Exploratory Literature Review of IEEE and Scopus Online Databases". In: *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*. Ed. by Tutut Herawan, Mustafa Mat Deris, and Jemal Abawajy. Springer Singapore, 2014, pp. 445–452.

[6] Jesper Kjeldskov and Connor Graham. "A review of mobile HCI research methods". In: *International Conference on Mobile Human-Computer Interaction*. Springer. 2003, pp. 317–335.

[7] Rodrigo Lozano et al. "A review of commitment and implementation of sustainable development in higher education: results from a worldwide survey". In: *Journal of Cleaner Production* 108 (2015), pp. 1 –18.

[8] Ken Schwaber. *Agile project management with Scrum*. Microsoft press, 2004.

[9] Ken Schwaber. "Scrum development process". In: *Business object design and implementation*. Springer, 1997, pp. 117–134.

[10] Kelsey Vaughan et al. "Costs and cost-effectiveness of community health workers: evidence from a literature review". In: *Human Resources for Health* 13.1 (2015), p. 71.

[11] Robert K. Yin. *Case study research and applications: design and methods*. 6. ed. SAGE Publications, Inc, 2017. ISBN: 1506336167.

Appendix A

Articles

(1) Menan Rajasegaram, Nicolai Vork, and Søren Lønsman Larsen. Involvement of Multiple Stakeholders in Agile Software Development: An Exploratory Literature Review. *Department of Computer Science, Aalborg University*. Aalborg, 2018.
(2) Menan Rajasegaram, Nicolai Vork, and Søren Lønsman Larsen. A Survey on Involvement of Multiple Stakeholders in the Software Industry. *Department of Computer Science, Aalborg University*. Aalborg, 2018.
(3) Menan Rajasegaram, Nicolai Vork, and Søren Lønsman Larsen. Involvement of Multiple Stakeholders in Scrum: A Case Study on Prescription of Medication. *Department of Computer Science, Aalborg University*. Aalborg, 2018.

Involvement of Multiple Stakeholders in Agile Software Development: An Exploratory Literature Review

Menan Rajasegaram
Hobro, Denmark
mrajas13@student.aau.dk

Nicolai Vork
Aalborg, Denmark
nvork13@student.aau.dk

Søren L. Larsen
Aalborg, Denmark
slar12@student.aau.dk

ABSTRACT

In modern development the involvement of stakeholders have become more prevalent compared to the traditional development methods, like the waterfall method. Some projects requires multiple stakeholder involvement, which can be difficult, however agile methods such as XP and Scrum does not provide a clear guideline for involvement of multiple stakeholders. This paper reviews literature on how to involve multiple stakeholders in agile software development. We conducted an exploratory literature review. We reviewed 15 papers. These were found using the criteria that the title had to indicate is related to multiple stakeholder, and that the abstract should also reflect this. We then used an affinity diagram to divide the articles into HCD in Scrum, Communication in Agile Development, Requirements Elicitation, and Project Decision Making. In the analysis we find that inclusion of multiple stakeholders does not have one well defined method. According to the articles the stakeholders should be involved early, in order to ensure misunderstandings are corrected early. Further to ensure the product is usable for the stakeholders. There should be a strategy for the involvement, this could be having the stakeholders discuss amongst themselves and the developers to find a solution that satisfies them all. Conflicts should be handled in a meaningful way, this could be by using framing to communicate the requirements to the stakeholders, in order to make them consider each others requirements. Inclusion of usability practices, such as having the stakeholders conduct usability tests to validate the product.

Author Keywords

multiple stakeholders; multi-stakeholder; exploratory literature review; agile software development; stakeholder involvement

INTRODUCTION

In modern software development, involvement of stakeholders has become more prevalent, with the rise of agile development methods [15] [25] [3]. In a survey conducted in 2017 [12], 71 % of 3234 respondents used agile methods in their projects.

The idea of agile software development was introduced in 2001 in the Agile Manifesto. One of the values of the manifesto is *customer collaboration over contract negotiation*. Principles behind the Agile Manifesto related to this value are customer satisfaction and welcoming changes [5]. Customer satisfaction is about satisfying the customer by delivering valuable software throughout development, and welcoming changes is about accepting changing requirements, even late in the development, such that it increases the customer's competitive advantage.

In agile methods such as Scrum, the Scrum Team achieves customer collaboration through collaboration with the customer in Sprint Planning and Sprint Reviews [22]. In Extreme Programming the value is achieved by always having the customer available to the development team during development [4].

Besides the customer, there may be other stakeholders in a project such as end users, suppliers, sponsors, and domain experts. However, involving multiple stakeholders in a project can be a difficult task, since the stakeholders can have different needs that conflict with each other [24, P.102].

In Scrum projects, key stakeholders are invited to Sprint Reviews, where they can review the developed increment from the Sprint. However, they are not included in Sprint Planning or during the development of the increment in each Sprint [22]. In addition, Scrum does not address how to determine which stakeholders are key stakeholders or how conflicts between these are resolved [22]. Furthermore, Extreme Programming does not have explicit support for involving other stakeholders than the customer in the development process [4].

This paper presents an exploratory literature review of how multiple stakeholders can be involved in agile software development.

The remaining sections are organised as the following: The next section presents related work. After this, we describe our research method. We then present the results of the literature review. Lastly, we discuss these results and conclude on the research question.

RELATED WORK

To find related literature reviews, we used the databases Google Scholar and Science Direct. The used keywords are ”literature review”, ”agile”, and ”stakeholder involvement”. We did not find any literature reviews on specifically involvement of multiple stakeholders in agile software development.

Instead we reviewed other literature reviews, which are about agile software development or involvement of multiple stakeholders in specific phases of agile software development.

Brhel et al. [6] conducted a systematic literature review on user-centered agile software development. They constructed principles for user-centered agile software development. They identified 83 relevant papers and analysed these. Five principles were derived from the review. One principle derived from the review was *continuous stakeholder involvement*, which is about involving the stakeholders throughout design and development to receive feedback on the product. Another principle was *separate product discovery and product creation*, which is about having an up-front analysis and design of requirements and system interactions before implementing a product. They argue that by having up-front analysis and design before implementation, the product can be both useful and usable.

A study conducted by Inayat et al. [11] discusses challenges in agile requirements engineering. They performed a systematic literature review on agile requirements engineering. Their focus was to find practices used in agile requirements engineering, challenges of traditional requirements engineering that are resolved by agile requirements engineering, and challenges of agile requirements engineering. The literature review resulted in an analysis of 21 papers. They identified practices in agile requirements engineering such as *customer involvement and interaction*, which is about involving customer to define, clarify, and prioritise requirements. Another practice is *review meetings and acceptance tests*, which is about receiving feedback in meetings to validate the requirements. With acceptance tests the authors state that they *"increase team, customer and domain expert collaboration as well as reduce the severity of defects and regressions"* [11]. Further, they identified challenges of agile requirements engineering such as *customer inability and agreement*. Customer inability refers to the incompetence of a customer, while customer agreement concerns the disagreement between multiple customers involved in a project. This can affect the collaboration.

It is clear that existing literature reviews examine the areas of both involvement of multiple stakeholders and challenges in agile software development separately. However, there seems to be a gap in existing literature on how to involve multiple stakeholders during agile software development.

RESEARCH METHOD

Since existing literature on the subject of involvement of multiple stakeholders in agile software development is relatively limited, and we did not find any literature reviews on the subject, we conducted an exploratory literature review.

Kitchenham describes guidelines on how to conduct a systematic literature review [14]. Since these guidelines focus on how to conduct a systematic literature review rather than an exploratory one, we do not follow all the guidelines, but only those relevant for an exploratory literature review. Thus, our review process consisted of the following three phases: Planning, conducting, and reporting.

In the planning phase we determined the keywords, databases, and inclusion criteria. The list of keywords are given in Table 1.

Database	*Keywords*
Google Scholar	agile communication stakeholders
Google Scholar	agile stakeholders
Google Scholar	agile communication
Google Scholar	agile stakeholder involvement
Science direct	multiple stakeholders
Science direct	multi-stakeholder
Scopus	multiple stakeholder involvement
Scopus	multiple stakeholder involvement agile
Scopus	user involvement
Scopus	user involvement agile

Table 1. A table containing the keywords that were used to retrieve relevant papers.

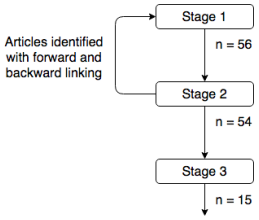


Figure 1. An image illustrating the stages and the number of papers (n) that satisfied the inclusion criterion in each stage.

The used databases were Google Scholar, Science Direct, and Scopus. The inclusion criteria are described in the conducting phase.

The conducting phase included three stages as illustrated on Figure 1. Further, the figure shows the number of papers that satisfied the inclusion criteria in the stages. In stage 1, we identified 56 papers in total after forward and backward linking of the papers that had passed the inclusion criterion in stage 2. In stage 2, we identified 54 papers, and in stage 3, 15 papers passed the inclusion criterion and were used for the review. A description of each stage are given in the following.

In stage 1 we entered the keywords in the databases. The inclusion criterion to get to the next stage was that the title had to indicate that the paper had to do with multiple stakeholders. We reviewed the titles of the papers on each page of the search results in the databases, until titles of the papers did not satisfy the inclusion criterion.

In stage 2 we read the abstract of each article that had passed the previous stage. The inclusion criterion in this stage was that if the abstract indicated a connection to multiple stakeholders, then the article would pass to stage 3. We performed backward and forward linking on articles that passed the inclusion criterion in stage 2. The articles that were found using backward and forward linking were also evaluated against the inclusion criteria of stage 1 and 2.

In stage 3 we skimmed each paper that had passed the previous stage. If the paper was relevant for the research question, then it was used in the reporting phase.

In the reporting phase we abstracted and generalised the results. In order to abstract and generalise the results of the

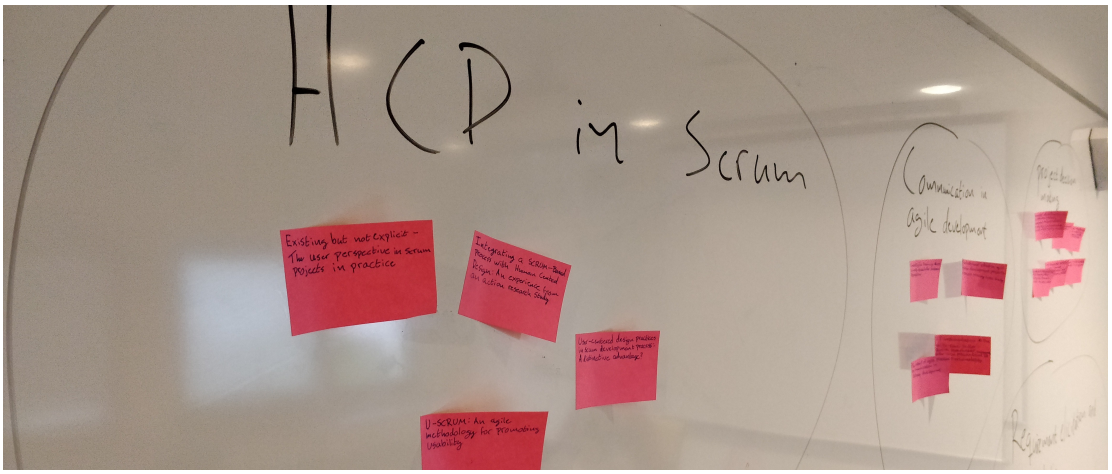


Figure 2. An affinity diagram showing post-it notes in the categories.

literature review, we used an affinity diagram [8]. As illustrated on Figure 2, we constructed an affinity diagram. This was achieved through the following steps: First, we read all articles to get an understanding of the literature. We then brainstormed and discussed different categories and used post-it notes with the title of each paper to categorise the papers in the discussed categories. Finally, we selected the following categories:

- HCD in Scrum
- Communication in Agile Development
- Requirements Elicitation
- Project Decision Making

These categories are described in the following section along with related papers from the literature review.

RESULTS

An overview of how the articles are categorised is in given in Table 2.

In the following sections each of the categories are explained, and the results gained from the exploratory literature review are written in the related categories. The last paragraph in each category summarises the presented articles.

HCD in Scrum

Human Centred Design in Scrum (HCD in Scrum) covers how to incorporate User Centred Design (UCD) and similar processes into Scrum.

In [27] they examine how the user perspective is affected by using Scrum in practice. This is done through interviews with 21 professionals. The interviewees are three Scrum managers, nine team members, five usability specialists, and four business specialists. When the interviewees were asked, who is responsible for the product being usable, they disagreed. For instance, six interviewees responded that the Scrum managers are responsible, while three of the interviewees responded that no one is responsible. This leads to the conclusion that it is unclear who is responsible for the usability of the product, what the usability goals are, and how to involve users and receive user feedback. Further, they conclude that collaboration with users is performed in informal ways. The article contributes in answering the research question by concluding there are limitations with incorporating usability and users in Scrum.

In [2] they focus on how to integrate HCD into Scrum. They use a case study as research method. They propose a variation of Scrum called HCD Scrum. HCD Scrum has a Customer Committee, Inception, Sprint n.0, SCRUM Islands, (IN)Sprint Review, and Project Retrospective, which differ from the traditional Scrum. Furthermore, the Sprints are limited to 1 week time boxed Sprints, and Sprint Retrospectives are removed from the Sprints. The Customer Committee consists of a Product Owner and at least two people from the customer side, who are preferably an end user and a business expert. They are involved in creating the Product and Sprint Backlogs, and involved in the Sprint Reviews. Inception is used to describe the initial phase, where stakeholders are interviewed, and field studies are conducted to elicit the initial requirements of the product. Sprint n.0 is added to design the architecture of the product and construct a high level prototype. HCD Scrum uses SCRUM Islands, which means the members of the development team are placed at a round table, ideally four people, to accommodate better communication flow between them. The table consist of the Product Owner from the Customer Committee, who is also a part of the development team, a graphic designer, and two software engineers. During the Sprint an (IN)Sprint Review is conducted, where the end users of the Customer Committee use the system and test it. The results are discussed during the Sprint Review. By doing this, the team achieves continuous validation. Project Retrospective replaces the Sprint Retrospectives. They argue that there is no need for a retrospective at the end of each Sprint, because the developers and stakeholders, through the Product Owner, are expected to communicate. However, to ensure the development process is reflected on, a Project Retrospective is

Article	Communication in Agile Development	HCD in Scrum	Project Decision Making	Requirements Elicitation
[27]		X		
[2]	x	X	x	x
[23]	x	X		
[1]		X		
[9]	X			
[13]	X			
[26]	X			
[19]	X			
[21]	x		x	X
[16]				X
[10]		x	X	
[17]			X	x
[7]	x		X	
[18]	x		X	
[20]	x		X	

Table 2. The selected articles and how they are categorised. The main category an article is about is marked with X, and subcategories are marked with x.

conducted at the end of the project. They recommend the use of short Sprints, meetings, and focus groups, which help the stakeholders understand the requirements and limit the impact of misunderstandings. The article contributes in answering the research question, since they make recommendations on how to involve stakeholders.

[23] describes a modified version of Scrum called U-SCRUM, which incorporates usability into the development. They conduct case studies of both Scrum and U-SCRUM used in practice and base their findings on these. U-SCRUM has two Product Owners, where one focuses on the traditional tasks in Scrum, and the other focuses on usability and user experience. During the development personas are used to understand how the user would use the product. These are introduced to the team, and then physically posted around the work area, such that the development team keeps focus on the users of the product. U-SCRUM adds an explicit User Experience Vision that is used to communicate the product to the stakeholders. The vision acts as a basis for discussions and helps stakeholders communicate improvements. They conclude that U-SCRUM can be used to construct a product of with improved usability and user experience. The article contributes in answering the research question by providing a development method for including the user perspective.

In [1] Scrum is modified to include user-centred design practices to improve the usability of the product. They use both a case study and observations as research methods. The modified version of Scrum divides development into three stages: Planning, design sprint, and development sprints. Planning involves identification of all stakeholders and elicitation of the initial requirements performed by Product Owner. To conduct the planning phase the development team can use methods such as task analysis, 1-on-1 interviews, and focus groups. After the planning phase, the design sprint is conducted, where the development team for instance uses paper prototyping and user stories to design the product. During this Sprint, they argue that it is very important to gather feedback from the users in order to validate the design. The development sprints are the last phase, where they complete the functional requirements and perform usability tests. Further, the product is redesigned based on results from usability tests in previous development sprints. Other than the established roles in Scrum, they have a usability commander, a technical leader, a test engineer, and UI Designers. They conclude that personas and user stories did not aid the development, but the inclusion of a usability commander and paper prototyping did. In addition, they conclude that customer satisfaction improves with the inclusion of user-centred design practices, and reduces the cost by minimising the amount of changing requirements. The usability commander and prototyping also ensure the users will be satisfied. The article contributes in answering the research question by incorporating user-centred design practices into Scrum and recommending methods that improves customer satisfaction.

Three of the articles in HCD in Scrum create variations of Scrum that incorporate certain stakeholders or perspectives in the development. This is done in different ways. For instance, [2] and [1] both use a design sprint to incorporate usability in the development. The articles presented in this category improve usability or user experience of the product by allowing the stakeholders to comment on usability or user experience. This indicates that Scrum has problems with integrating aspects of HCD such as usability and user experience.

Communication in Agile Development

This category covers communication between developers and stakeholders in agile development.

In [9] they construct a better way to frame tradespace (tradespace is a contraction of trade-off playspace) problems. When a speaker presents a solution to a listener, framing is about how the solution is presented. Certain aspects of the solution can be highlighted in the presentation, while others can be hidden. An aspect can also be presented in many different ways. Thus, the speaker can control, how the solution is perceived by the listener. When developers try to find the best solution to a problem together with multiple stakeholders, a Best Alternative To Negotiated Agreement (BATNA) can be constructed for each stakeholder. A BATNA can be used as a reference to negotiate from. Developers can present BATNAs

to stakeholders and frame them in such a way that the stakeholders can reach an agreement. The proposed solution to frame tradespace problems is not tested, but only described in detail and reasoned about. The article contributes in answering the research question by providing an understanding of how to present solutions and ideas to multiple stakeholders in a way that the stakeholders understand each other's ideas and opinions and help them reach an agreement.

In [13] they describe the value of engagement of stakeholders in the development of IT systems. In order to do this, they conducted a case study. They reviewed the effects of the Informing Healthcare programme in Wales, where they found that the inclusion of both healthcare personal and patients are important. Furthermore, they found that involvement of stakeholders improve the quality of the product. They recommend that the involvement should begin early and be consistent throughout the development process. Further, it is important that the involvement is positive and open with no special treatment of any stakeholders. They conclude that this will improve usability of the product and also lower the cost of the product. The article contributes in answering the research question by providing guidelines for involvement of stakeholders, such as involving the stakeholders early and throughout the development.

In [26] they analyse how agile development methods and the participation of customers and users influences information systems development projects. They used case study as research method. They report from the case study that it became clear during development that the developers were not able to deliver 50 % of the highest prioritised requirements. They analysed why this problem occurred with regard to e.g. customer maturity and communication levels. The customer maturity is their ability to provide sufficient support to the development team, lack of maturity can lead to lack of participation and engagement. The communication level can be described in terms of how many has access to communication, how rich the communication is, and whether it is one-way or two-way communication. A high communication level can alleviate other problems in the development, such as the maturity level. They conclude that the involvement of stakeholders with weak competences can be alleviated with communication. Communication with stakeholders can also be used to reach a mutual understanding of the project and keep the stakeholders involved in the project. They recommend the use of an agile development method with continuous delivery to improve the communication through the deliverables. Further, they recommend the use of an expert consultant as the Product Owner, which helps the development team understand the needs of the stakeholders. Finally, they recommend having the customer assign local domain experts to the project. The article contributes in answering the research question by recommending how to involve stakeholders.

In [19] they analyse communication in agile software development, and how it affects development. They do this by conducting two case studies, which are both agile development projects, and comparing the communication in them. They found that the communication techniques with a positive influence on communication are open office space, daily meetings, story or task boards, iteration planning, reflection workshops, pair programming, and continuous integration. They conclude that the communication with external stakeholders is not improved solely by the use of agile development methods, but also requires the communication to follow the means of communication in the agile development method. This further requires an understanding of the method by both the development team and the stakeholders. This understanding is achieved through educating the stakeholders in the development method. The article contributes in answering the research question by stating that the use of an agile development method is not enough to ensure the communication with multiple stakeholders is effective.

In the articles [9] and [19] they state that in order to improve communication with stakeholders, it is important that communication is framed in a constructive way which make stakeholders more likely to reach consensus, and that communication is not effective just by using an agile development method, respectively. Further, in [13] they conclude that stakeholders should be involved early and throughout development, which reduces cost and improves usability. They also involve two stakeholders, namely patients and healthcare personal with similar recommendations to [10], which indicates that the lessons learned from involving one stakeholder can be applied for multiple stakeholders as well. Generally, articles in this category recommend direct involvement of stakeholders to improve communication.

Requirements Elicitation

This category is about the process of eliciting requirements for a project and how to manage requirements.

In [21] they gamify the process of requirements elicitation to ensure most or all requirements are gathered. This is done through two case studies. In the first case study they construct a physical board game with seven stakeholders as players. In the second case study they construct a web-based prototype of the game with 17 stakeholders as players. The game is based on the six thinking hats, where each hat has a specific colour and a way of thinking, such that when figuratively wearing a hat, facts and comments should follow the hat's way of thinking. The blue hat is interested in understanding, what the topic is. The white hat requests the exposure of facts and numbers objectively. The red hat reacts to comments in an emotional way without justification. The black hat applies logic to understand why to be conservative. The yellow hat uses logic to understand why to be optimistic. The green hat uses provocation and exploration to be creative. A game moderator or product manager sets up the initial requirements and categorises them. These categories are then used for elicited requirements during the game. The players then take turns coming up with new requirements. The other players then comment on the requirements or rate them. The comments and ratings are based on a specific hat, which they choose themselves. A player scores points by expressing new requirements, comments or ratings. The comments must be constructive, such that the game is productive. After the game has finished, the project manager resolves conflicting opinions between stakeholders. They found that the game helped with

requirements elicitation and involved the stakeholders more in the process. The project manager found the input helpful for constructing a solution that is beneficial to all stakeholders. The article contributes in answering the research question by providing a gamified method for requirements elicitation with multiple stakeholders. The method demonstrates that the stakeholders do not need to be involved directly in the final requirements selection.

In [16] they construct a web tool for prioritisation of requirements. The web tool uses a social network of stakeholders, where the stakeholders can comment on the requirements and rate them on a scale from 1 to 5. The features are identification of requirements, prioritisation of requirements, recommendation of requirements, and highlighting of conflictual requirements among stakeholders. They conclude that the tool is useful and powerful, which they base on early trials. The article contributes in answering the research question by constructing a tool for identification of stakeholders and prioritisation their requirements.

In general the articles suggest that the stakeholders should consider each other's opinions and make decisions based on it. [21] and [16] present ways to elicit requirements. [21] presents a game that helps to elicit requirements, and [16] presents a tool that uses a social network of stakeholders to identify and prioritise requirements. Both of these only use stakeholders in requirements elicitation and not in requirements selection.

Project Decision Making

This category is about reaching consensus among multiple stakeholders through the use of development methods or mathematical models.

In [10] they conduct a case study of a firm moving from traditional development to agile development using Evolutionary Project Management (EVO) as development method. The development method is used to create a product in increments, which leads to several versions of a product. The stakeholders provide measurable goals that the developers seek to meet. Further, the stakeholders test the versions and provide feedback on these. This requires that the stakeholders maintain engagement throughout development, as they are an important part of the development method. The developers found continuous integration helpful, which is not a requirement of EVO, to ensure a working version of the product at all points of development. The involved stakeholders have a close relationship with the developers and are expert users, which helped to keep them engaged in the development. They conclude that there are prerequisites for using agile methods, which are that stakeholders should only be involved when relevant, and that stakeholders should be managed proactively. They also conclude that the use of stakeholders for testing versions of the product and setting measurable goals throughout the development contributes to a better product. According to [10], there are three benefits of using EVO. The first benefit is that direct involvement of stakeholders is motivating for the developers. The second benefit is that developers are more confident that they develop a product that satisfies the needs of the stakeholders. The final benefit is that transparency of EVO is increased for developers and stakeholders. The article contributes in answering the research question by involving multiple stakeholders in an agile development method and highlighting the benefits.

In [17] a framework for optimisation of environmental management problems is constructed. First the framework is constructed after which a case study is used to test it. The framework incorporates several stakeholders and objectives. The framework works by first having each stakeholder optimise and analyse the possible solutions and then select a BATNA. The BATNA selected should be their Pareto optimal solution, which represents a solution, where no aspect of the solution can be improved without making another aspect of the solution worse. The stakeholders share their preferred solutions, which are compared to the other stakeholders' BATNA, and they discuss which objectives they are willing to lower the value of. They then negotiate and identify cross-stakeholder efficient solutions, after which one or two solutions should be identified for further analysis. Similar solutions are compared to ensure no better solution exists. If there are no better solutions, then the solution is optimised to a Pareto optimal solution. They conclude that their framework only finds a near optimal solution, but the framework is still useful to highlight disagreements among stakeholders and help stakeholders choose a solution. The article contributes in answering the research question by presenting a framework for finding near optimal solutions for multiple stakeholders and objectives.

In [7], they make a Decision Making Procedure supported by mathematics to reach consensus in groups. The framework works in six steps, where the first step called step 0 is a preparation step. In the preparation step they set the number of iterations the process should run for, and when the group is considered to having reached consensus, such as 67% agreeing. The next step the group discuss the problem, and opinions are given with a supporting explanation. Then, in step 2, the individuals finds their preference vectors, which indicates how important they feel objectives are in numeric values. In step 3, the group members are presented with the group mean preference vector, and how close their preference vector is to it. Then it is tested against conditions set in the preparation step. If the group has reached consensus or if all iterations have run, the process terminates, otherwise the group will discuss and negotiate based on the data provided by the iteration, aided by a facilitator. Further, they repeat from step 2. While still at the conceptional stage, the authors present rationale for the solution, such as limiting the opportunities for scheming. The article contributes in answering the research question by providing a method for reaching consensus, which could be used throughout a development process.

[18] constructs a method for solving decision making problems with multiple stakeholders. The method is constructed and then demonstrated on a case study. Weights are assigned to the stakeholders. However, instead of assigning the weights according to how important the stakeholders are, the weights are assigned based on how the stakeholders rate alternatives. Stakeholders, who show a high difference in ratings, are given a smaller weight than those with lower difference. This is done to counteract any stakeholder from gaming the system. They

conclude that setting up alternatives for the stakeholders and then assigning them weights based on their ratings is a better solution than only assigning weights based on a perceived ranking. Further, they argue that it is easier to reach consensus with this method than using perceived rankings. The article contributes in answering the research question by highlighting the problem that stakeholders sometimes try to manipulate the outcome to their advantage in decision making.

[20] conducts a case study, where a firm transitions from traditional development to agile development. Stakeholders are divided into two tiers, where the first tier is customers, employees among others, and the second tier is stakeholder such as government and competitors. The article states that many agile adoptions fail or are made more problematic to adopt due to not considering all stakeholders. They present an idea of how to consider all stakeholders when transitioning to agile development. They divide the stakeholders into six primary stakeholder groups: Product Owner Team, Product Delivery Team, Program Sponsor Team, Product Consumers, Product Council, and Program Core Team. These stakeholder groups are involved in the different phases of the development depending on how relevant they are to involve in the given phase. The Product Owner Team is composed of anyone with a stake in the end product, who provide the Product Delivery Team with inputs about the product. The Product Delivery Team is composed of anyone with a stake in the development of the product. The Program Sponsor Team is composed of anyone with a stake in how the product fits in the product development organisation. Product Consumers is composed of anyone who has a stake in buying, using, or evaluating the end product. Product Council is composed of stakeholders, who are responsible for several products which form a portfolio such as a manager for several Product Owners. Program Core Team consists of stakeholders who facilitate general program and management activities. They conclude that dividing the stakeholders into the groups helps them to understand, when the views of a certain stakeholder should be considered. The article contributes in answering the research question by providing the understanding of why stakeholders are not equally important in all phases of development. It is also helpful for understanding when a group of stakeholders should be involved and how to involve the stakeholder, as not every stakeholder will benefit from participating in e.g. requirements elicitation.

In [10] they state that there are two prerequisites for the use of agile methods: Proactive management of the stakeholders and relevant involvement of stakeholders. In addition, they state that communication with stakeholders is a key factor to the success of development projects. Both [17] and [7] opt for mathematical approaches that find a Pareto optimal solution and preference vectors respectively. The framework presented in [7] uses the input from the stakeholders to reach consensus, which correlates to the other articles, which state that stakeholders should be included, when decisions are made. The process of reaching consensus is aided by a facilitator, which correlates to [21] in which a facilitator is also used. Generally, the articles suggest having a close relationship to the stakeholders in order to keep them engaged. Since all articles in this category involve stakeholders in the decision making process, it is clear that the stakeholders should be involved.

DISCUSSION

In this section we discuss involvement of multiple stakeholders in Scrum. We then discuss the inclusion criterion used in stage 1 of our review process. We also relate our findings to the related work. Finally, we present implications for practice.

The literature review revealed that there are still improvements that can be made on how to involve multiple stakeholders in agile development methods. Three of the articles in HCD in Scrum constructed different variations of Scrum that allow for stakeholder involvement during development. For instance, in [2] they include a Customer Committee that consist of a Product Owner and at least two other individuals from the customer side. During the Sprints some individuals from the Customer Committee are involved to test the system. Another paper [1] modifies Scrum to divide the development into planning, design sprint, and development sprint, since they argue that Scrum needs to involve stakeholders such that the development team can elicit requirements and gather feedback on the design. This indicates that there is a need for changes to Scrum that allows for involvement of stakeholders.

The inclusion criterion in stage 1 only required the titles of the found articles to indicate that they concerned multiple stakeholders. In retrospect, the inclusion criterion should also require keywords of the articles to indicate that the articles concerned multiple stakeholders, since a title may not cover every subject under research in an article.

Further in [6] they argue that product discovery and product creation should be separated. [1] did a similar separation, where they divided the development into planning, design sprint, and development sprint, such that identification of stakeholders, elicitation of requirements, and the design of the product is separated from the development. In [11] from related work, they found that there are challenges with requirements engineering such as customer incompetence and conflicts among the stakeholders. This correlates with [26], where they state that these challenges can be solved with improved communication. Further, they recommend the use of domain experts to overcome the customer incompetence. The challenge of conflicts between stakeholders is also handled in several articles in the category Project Decision Making, where they use different methods to handle the challenge. In [9] they also state that conflicts among stakeholders can be solved through framing, which can help the stakeholders understand each other's needs and wishes.

The results have two implications for practice. Involvement of stakeholders should be done as early as possible, e.g. to prevent misunderstandings between developers and stakeholders. Furthermore, usability of the product should be considered, as this can affect both the cost and quality of the final product.

CONCLUSION

This paper presented an exploratory literature review of how multiple stakeholders can be involved in agile software devel-

opment. We initially identified 56 papers. Then we sorted out papers that were not relevant for the research question using inclusion criteria. In total, 15 paper passed all inclusion criteria and were used for the review.

Based on the articles we conclude the following:

- There should be a supporting structure for involvement of stakeholders
- Stakeholders should be involved early and throughout the development
- Conflicts among stakeholders should be handled

Several articles proposed variations to Scrum to incorporate stakeholders and their perspectives. Thus, a supporting structure is needed for the involvement of stakeholders.

Some articles suggest early stakeholder involvement, since it has the potential to reduce the cost. Further, stakeholders should be involved throughout the development to continuously validate the product, which also potentially reduce the cost.

Several articles suggest handling conflicts among stakeholders. Different ways to do this are presented. Some articles suggest resolving conflicts through means of communication, while others suggest mathematical models for finding solutions that satisfy most or all stakeholders.

A limitation of this article is that we cannot guarantee that all relevant literature has been found, since we conducted an exploratory literature review. Another limitation is that we in stage 1 only reviewed, whether the titles of the found papers referred to multiple stakeholders. This adds to the fact that we cannot guarantee to have found all relevant literature.

Future work should focus on conducting a systematic literature review of the subject.

REFERENCES

1. S. Anwar, Y. H. Motla, Y. Siddiq, S. Asghar, M. Shabbir Hassan, and Z. I. Khan. 2014. User-centered design practices in scrum development process: A distinctive advantage?. In *17th IEEE International Multi Topic Conference 2014*. 161–166.
2. C. Ardito, M. T. Baldassarre, D. Caivano, and R. Lanzilotti. 2017. Integrating a SCRUM-Based Process with Human Centred Design: An Experience from an Action Research Study. In *2017 IEEE/ACM 5th International Workshop on Conducting Empirical Studies in Industry (CESI)*. 2–8.
3. M. A. Awad. 2008. A Comparison between Agile and Traditional Software Development Methodologies. `http://pds10.egloos.com/pds/200808/13/85/A_comparision_between_Agile_and_Traditional_SW_development_methodologies.pdf`. (2008).
4. Kent Beck. 1999. Manifesto for agile software development. (1999).
5. Kent Beck, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, and others. 2001. Manifesto for agile software development. (2001).
6. Manuel Brhel, Hendrik Meth, Alexander Maedche, and Karl Werder. 2015. Exploring principles of user-centered agile software development: A literature review. *Elsevier* 61 (2015).
7. Noel Bryson. 1996. Group decision-making and the analytic hierarchy process: Exploring the consensus-relevant information content. *Computers & Operations Research* 23, 1 (1996), 27 – 35.
8. Su Mi Dahlgaard-Park (Ed.). 2015. *Affinity Diagram*.
9. Matthew E. Fitzgerald and Adam M. Ross. 2014. Controlling for Framing Effects in Multi-stakeholder Tradespace Exploration. *Procedia Computer Science* 28 (2014), 412 – 421.
10. Geir Kjetil Hanssen and Tor Erlend Fægri. 2006. Agile Customer Engagement: a Longitudinal Qualitative Case Study. In *Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering*. ISESE, 164–173.
11. Irum Inayat, Siti Salwah Salim, Sabrina Marczak, Maya Daneva, and Shahaboddin Shamshirband. 2014. A systematic literature review on agile requirements engineering practices and challenges. *Elsevier* 51 (2014).
12. Project Management Institute. 2017. Success Rates Rise. `https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2017.pdf`. (2017).
13. Rasa Juciute. 2009. ICT implementation in the health-care sector: effective stakeholders' engagement as the main precondition of change sustainability. *AI & SOCIETY* 23, 1 (2009), 131–137.
14. Barbara Kitchenham and Stuart Charters. 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering. (2007).
15. Yu Beng Leau, Wooi Khong Loo, Wai Yip Tham, and Soo Fun Tan. 2012. Software Development Life Cycle AGILE vs Traditional Approaches. `https://pdfs.semanticscholar.org/69b1/9ddc8a578f4c63d1dfe15252a465ee12fe5d.pdf`. (2012).
16. S. L. Lim, D. Damian, and A. Finkelstein. 2011. StakeSource2.0: using social networks of stakeholders to identify and prioritise requirements. In *2011 33rd International Conference on Software Engineering (ICSE)*. 1022–1024.
17. Michael Di Matteo, Graeme C. Dandy, and Holger R. Maier. 2017. A multi-stakeholder portfolio optimization framework applied to stormwater best management practice (BMP) selection. *Environmental Modelling & Software* 97 (2017), 16–31.

18. Hojat Mianabadi, Abbas Afshar, and Mahdi Zarghami. 2011. Intelligent multi-stakeholder environmental management. *Expert Systems with Applications* 38, 1 (2011), 862 – 866.

19. M. Pikkarainen, J. Haikara, O. Salo, P. Abrahamsson, and J. Still. 2008. The impact of agile practices on communication in software development. *Empirical Software Engineering* 13, 3 (2008), 303–337.

20. K. Power. 2010. Stakeholder Identification in Agile Software Product Development Organizations: A Model for Understanding Who and What Really Counts. *AGILE Conference, 2010* (2010), 87 – 94.

21. Claudia Ribeiro, Carla Farinha, João Pereira, and Miguel Mira da Silva. 2014. Gamifying requirement elicitation: Practical implications and outcomes in improving stakeholders collaboration. *Entertainment Computing* 5, 4 (2014), 335 – 345.

22. Ken Schwaber and Jeff Sutherland. 2017. The Scrum Guide. `https://www.scrumguides.org/scrum-guide.html`. (2017).

23. M. Singh. 2008. U-SCRUM: An Agile Methodology for Promoting Usability. In *Agile 2008 Conference*. 555–560.

24. Ian Sommerville. 2010. *Software engineering*. New York: Addison-Wesley.

25. Marian Stoica, Marinela Mircea, and Bogdan Ghilic Micu. 2013. Software Development: Agile vs. Traditional. *Informatica Economica* 17 (2013), 64–76.

26. Bjørnar Tessem. 2017. The customer effect in agile system development projects. A process tracing case study. *Procedia Computer Science* 121 (2017), 244 – 251.

27. Åsa Cajander, Marta Larusdottir, and Jan Gulliksen. 2013. Existing but Not Explicit - The User Perspective in Scrum Projects in Practice. In *INTERACT*.

A Survey on Involvement of Multiple Stakeholders in the Software Industry

Menan Rajasegaram
Hobro, Denmark
mrajas13@student.aau.dk

Nicolai Vork
Aalborg, Denmark
nvork13@student.aau.dk

Søren L. Larsen
Aalborg, Denmark
slar12@student.aau.dk

ABSTRACT

When multiple stakeholders are involved during development, it can be a difficult task for the development team, since the stakeholders can have conflicting perspectives. This paper aims to present a survey on how the software industry involves multiple stakeholders during development and the challenges that the industry experiences. Initially, we conducted semi-structured interviews to get an understanding of how software companies deal with multiple stakeholders. We then created an initial web questionnaire. Further, we did pilot of the initial questionnaire. Based on the feedback of the pilot, we revised the questionnaire and distributed on 12 forums. We received 31 complete responses and 30 non-complete responses. Based on the responses, it was clear that the respondents use different methods for involving multiple stakeholders. The most used method for involving multiple stakeholders was meetings. Further, the most common challenge when involving multiple stakeholders was conflicting requirements. Lastly, the respondents accommodates the challenges differently, for instance through workshops and refinement meetings.

Author Keywords

multiple stakeholders; multi-stakeholder; survey; questionnaire; software development; involvement of stakeholders

INTRODUCTION

Since the introduction of the Agile Manifesto in 2001 [2], many new software development methods such as Scrum, Kanban, and Extreme Programming have emerged. Common for the development methods is that they are inspired by some or all of the four key values in the Agile Manifesto.

One of the key values in the Agile Manifesto is *customer collaboration over contract negotiation*, which encourages developers to collaborate with and involve customers during the development process [2]. An agile development method that draws from this value is Extreme Programming, where an on-site customer is at the development team's disposal throughout the development allowing for close collaboration between the developers and the customer [1].

In addition to the customer, who is involved in development methods such as Extreme Programming, there are other stakeholders as well such as users, sponsors, suppliers, and domain experts that are relevant to involve during development depending on the circumstances. Involving these stakeholders in addition to the customer can benefit the overall development. For instance, it is beneficial to involve users as stakeholders in the development, since it can lead to more accurate user-related requirements as well as the avoidance of implementing costly features that are either not wanted by users or unusable [4].

To involve and collaborate with multiple stakeholders in a development project can be a difficult task, since different stakeholders have different perspectives, which may conflict with each other [9, P.102]. Furthermore, stakeholders often do not know their own needs and therefore have trouble expressing them to developers [9, P.102]. They also use their own terms to communicate their needs and wishes, which may require implicit knowledge to understand [9, P.102], which further complicates the communication between stakeholders and developers.

This paper presents a survey on how the software industry involves multiple stakeholders during development, and the challenges that the industry experiences.

The rest of this paper is structured as follows: In the next section we examine related surveys to our research question. We then describe how we conducted our survey. After this, we present the results of the survey. Finally, we discuss the results and conclude on the research question.

RELATED WORK

To find related papers on involvement of multiple stakeholders during development, we entered the keywords "involvement", "multiple", "survey", and "stakeholders" into the databases Google Scholar and Scopus. The found papers did not specifically address involvement of multiple stakeholders, but areas related to it. In the following we describe these papers.

Gulliksen et al. [3] conducted a survey study on the usability profession of the Swedish software industry. The survey was conducted through a questionnaire, which was answered by 194 respondents. The focus of the study was to examine how the industry deals with usability. They reported that 62 % of the respondents indicated that they were not satisfied with the

Example of Provided Software	Number of Employees	Location	Occupation
Education platform	More than 800	Denmark	Head of Department
Healthcare application	Around 220	Denmark	Program Leader
Voting application	174	Denmark	Project Manager
Webshop	9	Denmark	Developer
Music application	5	India	Solution Architect

Table 1: Overview of companies that participated in the semi-structured interviews.

Example of Provided Software	Number of Employees	Location	Occupation
Public service system	More than 1700	Denmark	IT Consultant
Insurance system	Around 180	Denmark	Manager
Booking application	2	Denmark	Developer

Table 2: Overview of companies that participated in the pilot of the initial questionnaire.

amount of time spent on user involvement. The remaining 38 % reported that they spent a fair amount of time on user involvement. The survey also showed that users were involved the most during the testing phase, but the least during the implementation phase.

A survey study on user involvement in the Icelandic software industry was conducted by Larusdottir et al. [6]. The aim of the survey was to determine the perceived importance of usability and methods used to involve users. The survey consisted of a questionnaire, and it received 82 responses. They reported that adopters of Scrum perceived usability as less contributing to the success of software than adopters of other development methods. Further, the study showed that the most used method for involving users by software developers was meetings with users. Conversely, other methods such as questionnaires or surveys were much less used.

Kujala et al. [5] also performed a survey study on user involvement, but in the Finnish software industry. They focused on how user involvement affects the quality of requirements as well as project success. The survey consisted of both a questionnaire, of which there were 18 respondents, and 8 semi-structured interviews. The results of the survey showed that 11 of the respondents had direct user contact. Some of the user involvement methods used were customer visits and workshops, inspection, and participation. Further, they reported that direct contact with users had a positive effect on the quality of most of the requirements, especially those that were user related. They also reported that direct contact with users positively affected the perceived project success. Finally, they conclude from the results that some of the challenges in terms of requirements elicitation were lack of resources and not involving the customer or users sufficiently.

Based on these survey studies, we find that neither of them study specifically how the involvement of multiple stakeholders in a software project impacts development. Thus, there seems to be a need for research on how the software industry involves multiple stakeholders throughout development, and the challenges that follow.

RESEARCH METHOD

In order to examine how the software industry involves multiple stakeholders and the challenges it experiences, we conducted a survey of the software industry. The target respondents were individuals from the software industry around the globe. The survey consisted of four steps:

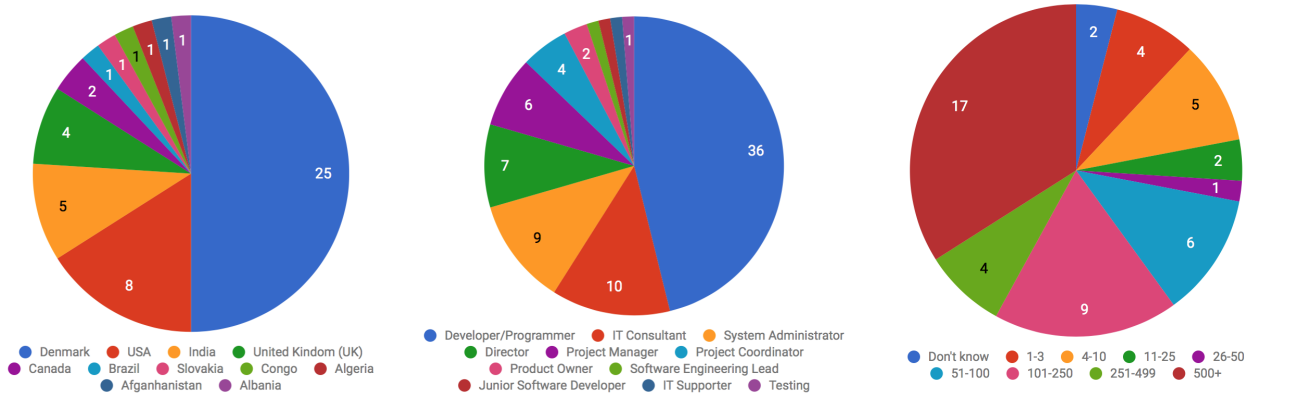
1. Semi-structured interviews
2. Pilot of initial questionnaire
3. Distribution of revised questionnaire
4. Analysis of responses

We conducted five semi-structured interviews with five different software companies to gain an initial understanding of how software companies deal with multiple stakeholders. Four of the five software companies were located in Denmark, while the last was located in India. We chose these companies, because they differ in the software they provide, their sizes in terms of the number of employees, and location of the company. An overview of the participating companies is given in Table 1. To conduct the interviews, we used a set of questions as a guideline. Based on the answers from the interviews, we created an initial web questionnaire with questions that were expected to answer the research question. The initial web questionnaire was created with the survey tool SurveyXact [7].

We then piloted the questionnaire with three software companies, which were not interviewed prior to answering the questionnaire. An overview of these companies is shown in Table 2. The responses were used to revise the questions that were unclear and to discard questions that did not yield useful results in terms of answering the research question.

The revised questionnaire was distributed on 12 software relevant forums, which are given in Table 3. In addition, 12 software companies were contacted directly and asked, if they wanted to fill out the questionnaire. The revised questionnaire is shown in Appendix 6.

On the first page of the questionnaire, prior to answering the questionnaire, the respondents were promised anonymity,



(a) Locations of the respondents' companies.

(b) Occupations of the respondents.

(c) Number of employees in the respondents' companies.

Figure 1: Overview of the respondents' demographics.

Linus Tech Tips
Tom's Hardware
Crazy Engineers
Devshed
Codeguru
Reddit (r/computerscience)
Reddit (r/developer)
Hardware Test Portal
Science Forums
Dream.In.Code
Young Programmers (Facebook group)
Software Outsourcing (Facebook group)

Table 3: Overview of forums that the revised questionnaire was distributed on.

since anonymity does not deter them from revealing negative aspects of how their company develops software.

In total, 530 individuals from the software industry had been to the first page of the questionnaire. Out of the 530 individuals, 31 answered all questions, and 30 answered only some questions. Thus, the questionnaire was answered by 61 respondents.

The responses from 10 respondents were removed, since the respondents had only answered the questions on the first page about their demographics. Further, one respondent had provided a spam response, which was also removed. Consequently, 50 out of the 61 responses were analysed.

We concluded the survey by analysing the results. Initially, we used SurveyXact to transform the results into comma-separated values. We then created a program in Python that read the comma-separated values and highlighted responses. This allowed us to get an overview of specific answers to questions and possible connections between answers to different questions.

Once we had an overview of the results, we conducted a brainstorming session, where we discussed the results and categorised them. The categories were used to decide, which results were relevant to answer the research question. The found categories that were relevant are the following:

- Software Development Processes
- Involved Stakeholders
- Stakeholder Involvement during Phases
- Challenges of Involving Multiple Stakeholders
- Methods for Stakeholder Involvement during Phases
- Elicitation of Initial Requirements
- Validation of Requirements
- Involvement of Stakeholders without a Representative
- Size of Company and Elicitation of Initial Requirements

Demographics of Respondents

The demographics of the respondents is shown in Figure 1. The respondents were asked which country they worked in, which occupation they had, and the number of employees of the company they worked in. The distribution of countries is shown in Figure 1a. 25 out of the 50 respondents were from Denmark, and the remaining 25 were from other countries. Furthermore, the majority of respondents were developers (36) as seen in Figure 1b. Lastly, 17 respondents worked in a company that had more than 500 employees, which is the biggest group, as shown in Figure 1c.

RESULTS

In this section, we present the results from the survey.

Some of the questions have fewer responses than others, since they only appeared to respondents, if they answered a certain way that would make certain questions relevant. Further, some questions have more responses than others, since they are multiple choice questions.

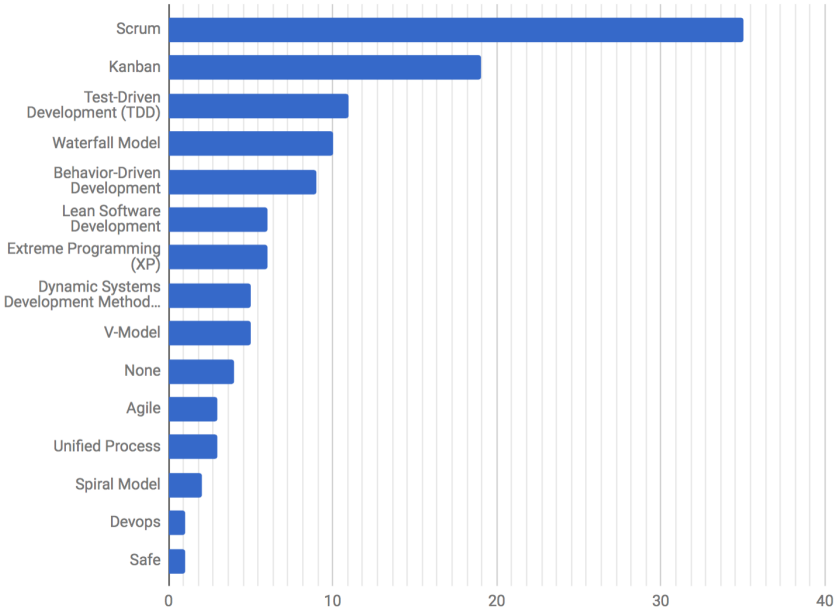


Figure 2: Distribution of used development methods.

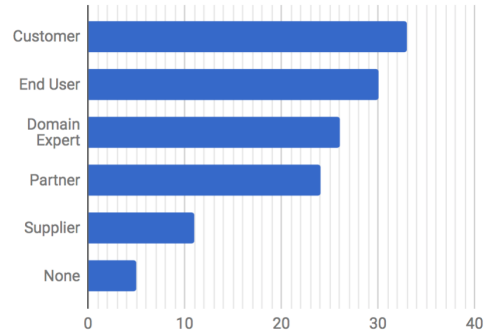


Figure 3: Involved stakeholders during development.

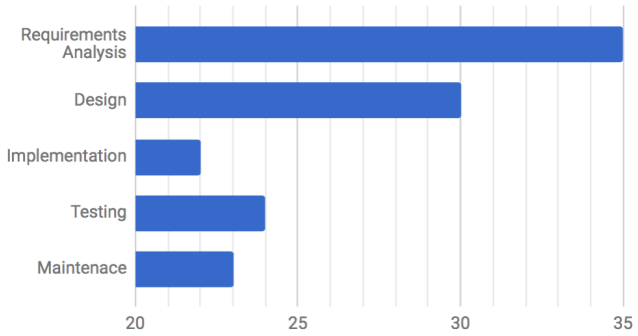


Figure 4: Phases showing when stakeholders are involved.

Software Development Processes

We asked the respondents, which development methods they used, and their answers are shown in Figure 2. 35 out of 50 respondents use Scrum, which is the most used development method among our respondents. One respondent noted that the company uses different development methods for each project. Further, all respondents who use more than one development method use at least one agile development method.

Involved Stakeholders

We asked respondents, which stakeholders they involved during development. The answers are shown in Figure 3. The three most involved stakeholders were customer answered by 33 out of 50 respondents, end user answered by 30 out of 50, and domain expert answered by 26 out of 50. 36 of the respondents answered that they involve two or more stakeholders, i.e. multiple stakeholders, while 9 respondents answered they only involve one. Only 5 of the respondents answered that they do not involve any stakeholders at all during development.

Stakeholder Involvement during Phases

We asked, when stakeholders were involved during development. The answers are presented in Figure 4. The most common phase to involve stakeholders is during requirements analysis answered by 35 out of 39 respondents. On the contrary, the least common phase is implementation answered by 22 out of 39 respondents. Less than half of the respondents (13) state that they involve stakeholders in every phase. Only 4 out of 39 respondents involved stakeholders in only one phase.

Challenges of Involving Multiple Stakeholders

We asked, which challenges the respondents face when involving multiple stakeholders. Figure 5 shows the respondents' answers.

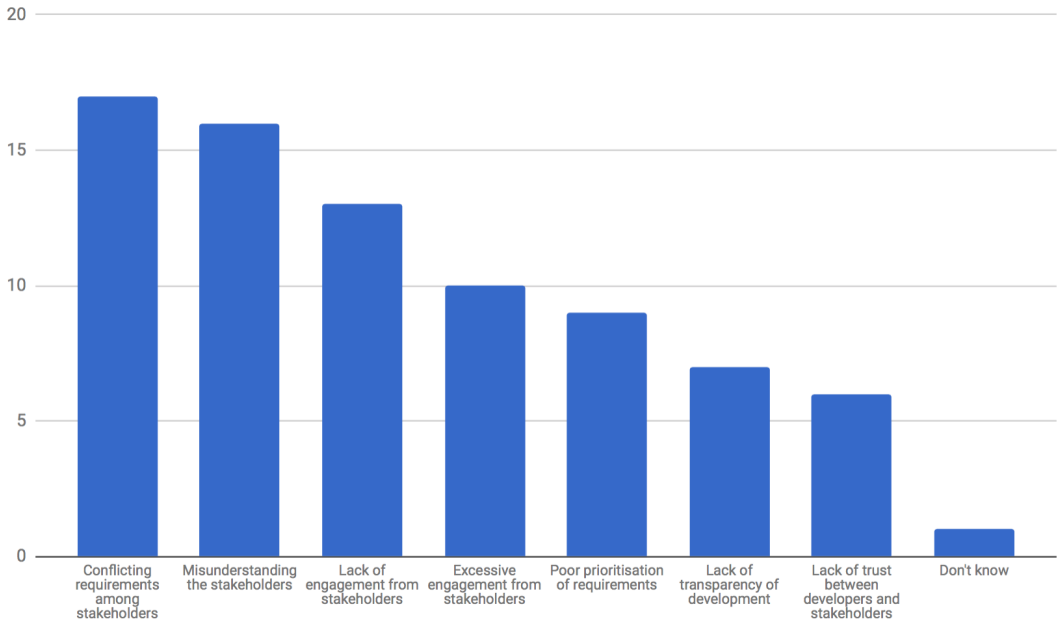


Figure 5: Challenges with involvement of multiple stakeholders.

The most common challenge faced by respondents is *conflicting requirements*, which 17 out of 30 respondents experienced. As a follow-up question, we asked how they handle the challenge, and one respondent stated that 95% of the system is designed with domain experts, and nothing further is done, since they *"can't make everyone happy"*. Another respondent stated that the stakeholders must sort it out on their own and provide a solution. A third respondent stated that they used refinement meetings. Finally, one respondent handled the challenge with workshops, where constraints and issues are presented and a solution is found.

The challenge *misunderstandings* was handled with meetings by two respondents. Another respondent handled the challenge through iterations on wireframes. One respondent simply stated that they clear up the misunderstanding. Another respondent stated that communication is the most important factor when developing software, and when a question arises, it is better to get it answered than to assume an answer. Finally, one respondent reports that they use experts or consultants to clear up misunderstandings.

Challenges with *lack of engagement* was handled by one respondent by arranging a meeting with the stakeholders. Another respondent stated that they would stop working on the requested work by stakeholders, until the stakeholders feel the need to engage them again. Finally, one respondent stated that they did nothing, since they do not have time to wait and would continue to work. When the system is finished, they would then charge stakeholders for any changes the stakeholders have.

The challenge *excessive engagement* was handled by one respondent by talking and sorting it out with the stakeholders.

The challenge *poor prioritisation of requirements* was handled by one respondent by arranging a meeting with the project manager concerning the prioritisation. Another respondent, who is a developer, stated that the challenge is not the developers' problem, and that they will simply develop whatever has the highest priority.

The challenge with *lack of trust* between developers and stakeholders was handled by one respondent by explaining to stakeholders that they must trust the abilities of the respondent, since the respondent is a technical expert.

There were no responses on how to handle the challenge with *lack of transparency*.

Methods for Stakeholder Involvement during Phases

We asked the respondents, how they involve stakeholders in different phases. The answers of the respondents are given in Table 4 and Table 5. In the requirements analysis phase, the most common method for stakeholder involvement is meetings answered by 21 out of 28 respondents. Two respondents elaborated on the question. The first respondent stated that they *"have individual interviews with end users to establish needs"* and then have meetings with other stakeholders. The second respondent answered that they ask, what the users want, and if the requirements specification is not sufficient, they hypothesise scenarios. They then present these to the stakeholders and ask them, what outcome they would expect in the scenarios.

In the design phase, the most common method is also meetings with the stakeholders answered by 16 out of 23 respondents. Two respondents elaborated on the question. The first respondent reported that they start with brainstorming sessions with the product and technical department, and then they iterate on

	[M]	[B]	[UC]	[DP]	[S]	[PP]	[I]	[W]	[P]	[FG]
Requirements Analysis	21	16	14	13	12	11	8	7	5	5
Design	16	13	10	13	10	11	8	4	4	3
Implementation	11	2	5	5	4	1	3	3	1	1
Testing	14	3	5	6	6	4	5	3	1	2
Maintenance	11	4	5	5	7	3	2	1	2	3

Table 4: Methods used for stakeholder involvement during the different phases. [M] : Meetings, [B] : Brainstorming, [UC] : Use Cases, [DP] : Digital Prototyping, [S] : Scenarios, [PP] : Paper Prototyping, [I] : Interviews, [W] : Workshops, [P] : Personas, [FG] : Focus Groups.

	[Q]	[PC]	[Su]	[Wi]	[FL]	[ST]	[AT]	[N]	[DK]
Requirements Analysis	4	1	1	0	0	0	0	0	0
Design	1	3	1	0	0	0	0	0	0
Implementation	1	1	2	0	1	0	0	0	0
Testing	1	1	2	0	0	1	1	0	1
Maintenance	2	1	1	0	0	0	0	1	1

Table 5: Methods used for stakeholder involvement during the different phases. [Q] : Questionnaires, [PC] : Phone Calls, [Su] : Surveys, [Wi] : Wireframes, [FL] : Feedback Loop, [ST] : Stakeholders Test Individually, [AT] : Acceptence Testing, [N] : None, [DK] : Don't know.

wireframes, until end users are satisfied, or until the deadline is reached. The second respondent repeated that they ask the users, what they want, and gaps in the requirements specification are presented as scenarios to the stakeholders. They then ask the stakeholders about the outcome, they expect from the scenarios.

In the implementation phase, meetings is still the most used method with 11 out of 16 respondents using it. The other methods for involvement all had 5 or less respondents stating that they use the methods.

In the testing phase, meetings with stakeholders is yet again the most common method of stakeholder involvement from the respondents with 14 out of 17 respondents using it.

In the maintenance phase, meetings is still the most common among respondents with 11 out of 17 respondents using it. One respondent answered that they do not use any methods for involvement, but elaborated that stakeholders are only involved if they want to change the solution.

Elicitation of Initial Requirements

We asked the respondents, how they elicit initial requirements. Figure 6 shows that 28 out of 39 respondents conduct meetings with stakeholders, which is the most common method. The second most used method is requirements specification provided by stakeholders, which 21 out of 39 respondents answered. Furthermore, 20 out of 39 respondents use their own knowledge to create initial requirements.

Validation of Requirements

We asked the respondents how they validate requirements. Figure 7 shows that the most common method for validation of requirements is acceptance testing, which 22 out of 29 respondents use. Three of the respondents do not validate with stakeholders at all, where one of these responded with *"looks good, ship it!"*.

	Focus Groups	Workshops
500+	6	7
251-499	1	2
101-250	1	2
51-100	0	0
26-50	0	0
11-25	0	0
4-10	0	0
1-3	0	0

Table 6: Number of employees in companies, and methods they use to elicit initial requirements.

Involvement of Stakeholders without a Representative

We asked the respondents how they handle stakeholders that do not have a representative. Only three respondents answered this question. One respondent stated that they use a chat forum, where the stakeholders can express their opinions. Another respondent stated that they approach a subset of the users. Finally, a respondent stated that they engage with multiple people within every customer company at all levels, which they state is key to their success.

Size of Company and Elicitation of Initial Requirements

Based on the answers from respondents, we found a connection between the number of employees in their companies and some of the methods used for elicitation of initial requirements. Table 6 shows this connection for the methods workshop and focus group. 8 out of 39 respondents answered that they use focus groups to elicit the initial requirements, and all their companies had more than 100 employees. The majority of these answers (6) were companies with more than 500 employees. Furthermore, 11 of the respondents answered that they use workshops, and their companies also had more than 100 employees. Most of these answers (7) were companies with more than 500 employees. No respondents, who were

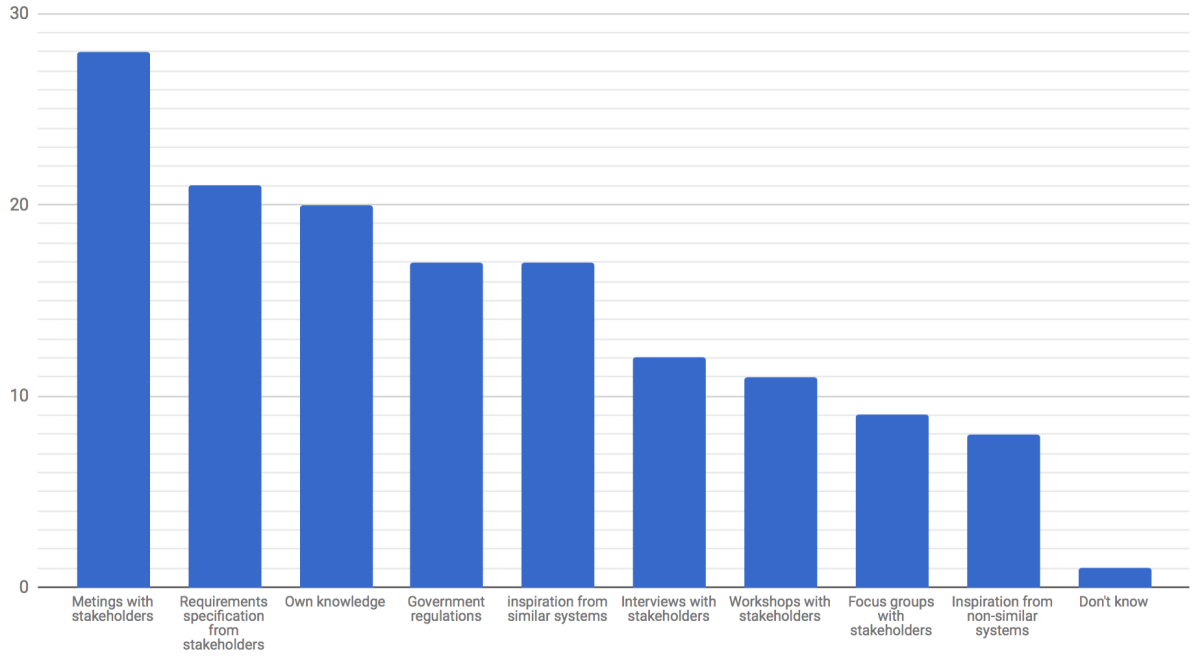


Figure 6: Methods used for elicitation of initial requirements.

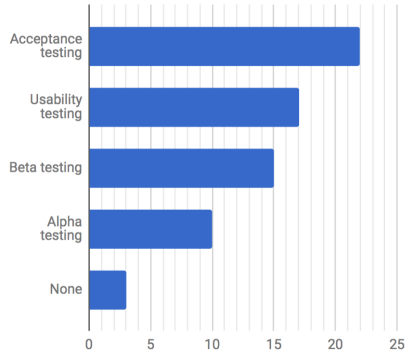


Figure 7: Methods used for validation of requirements.

employed in a company with less than 100 employees, used focus groups or workshops. These results indicate that smaller companies do not have enough employees, and therefore not the resources, to use focus groups or workshops to elicit the initial requirements.

DISCUSSION

In this section, we discuss the relevance of the research question. Further, we relate our results to the related work. We then discuss why we could not find more connections between answers to different questions. Finally, we discuss implications for practice.

Among our respondents most of them involve stakeholders in their development. One respondent stated that the way they handle multiple stakeholders is a key factor in their project success. Further, all but one of the respondents, who answered the question about which challenges there exists when involving multiple stakeholders, reported that there were at least one challenge with the involvement of multiple stakeholders. We also received few answers on how to handle the challenges experienced when involving multiple stakeholders, and these answers expressed different ways of handling the challenges. Finally, the most used method for involving multiple stakeholders experienced by the respondents was meetings. They did however not further elaborate on how they used the method. The research question is therefore relevant.

Related work has looked into user involvement. In one of the papers, it is concluded that the most used method for user involvement is meetings. This correlates with our findings, where the respondents answered that meetings was the most used method for stakeholder involvement in all phases of development and maintenance. Furthermore, another paper concludes that users are involved most during the testing phase, but the least in the implementation phase. This partially correlates with our findings, since we found that stakeholders were involved most during requirements analysis phase, and the least during the implementation phase. Our findings do not completely align with related work, but this can be explained by the fact that the related work examines only involvement of users as opposed to involvement of multiple stakeholders.

Our questionnaire was answered by 61 individuals, where 30 of these did not answer all the required questions. In total, however, the questionnaire was seen by 530 recipients, and thus 88.5 % of the the recipients did not answer any questions at all. A reason could be that the recipients do not have enough knowledge about the subject, and therefore refrain from answering the questionnaire. Another reason

could be that the recipients do not find the subject interesting enough to spend their time answering the questionnaire.

Apart from the connection between the size of the respondents' companies and the methods they use for eliciting initial requirements, we did not find other connections between the answers. There are two reasons to this. The first reason is that we only have 50 usable responses. Therefore, the numbers we get from comparing answers become too small, and we cannot tell with certainty, if there is a connection or not. The second reason is that some questions are multiple choice. For instance, the respondents could choose several development methods, if they use more than one in their company. Consequently, we cannot tell with certainty whether a given development method has a connection to another question, since most of the respondents (33 out of 50) answered they use several development methods. In retrospect, we could have forced respondents to only choose one option for certain questions. Thus, it would be easier to establish connections between the answers to these questions and other questions. However, the answers would not be completely accurate, and we would not get a complete overview of, for instance, which development methods the industry uses.

In the questionnaire the most common challenge with involvement of multiple stakeholders was conflicting requirements followed closely by misunderstandings. This indicates that in practice the stakeholders are not involved enough in the development, or at least that communication is a contributing factor to these challenges. 9 out of 30 respondents also indicated that one challenge faced was poor prioritisation of requirements. 7 out of 9 respondents who responded with poor prioritisation use Scrum, which indicates that there is a need for an improvement of the prioritisation process in Scrum. A reason for this could be that Scrum does not describe how to prioritise requirements [8].

CONCLUSION

This paper presented a survey on involvement of multiple stakeholders during development and the challenges experienced in the software industry. We received 61 responses of which 31 were complete responses, and 30 were non-complete responses. 50 responses were used in the results after removing 11 unusable responses.

Based on the results from the survey, we conclude the following:

- The industry uses mostly meetings to involve stakeholders
- The industry experiences challenges with involvement of multiple stakeholders
- The industry accommodates challenges differently

Among the respondents, the most used method for involvement of multiple stakeholders is meetings with stakeholders. This was the case for all phases of development and maintenance. Other popular methods are brainstorming and use cases.

The three most common challenges with involvement of multiple stakeholders encountered by the respondents were conflicting requirements, misunderstanding stakeholders, and lack of engagement from stakeholders. The rest of the challenges were experienced by 10 or less of the 30 respondents that answered the question.

The respondents accommodate the challenges they experience in different ways. For instance, one respondent reported that they use workshops to handle conflicting requirements, while another respondent reported they use refine meetings to handle the same challenge.

When gathering responses, we only received 50 usable answers. Consequently, it is difficult to generalise the results, especially on a global level. Further, some questions allowed respondents to choose several answers, and thus it is difficult to make connections between answers to different questions.

Future work could examine the reasoning behind the answers provided by individuals in the software industry. The reasons could help to highlight connections between answers to different questions, and also reveal new areas to research, which relate to involvement of multiple stakeholders.

REFERENCES

1. Kent Beck. 2000. *Extreme programming explained: embrace change*. addison-wesley professional.
2. Kent Beck, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, and others. 2001. Manifesto for agile software development. (2001).
3. Jan Gulliksen, Inger Boivie, Jenny Persson, Anders Hektor, and Lena Herulf. 2004. Making a difference: a survey of the usability profession in Sweden. In *Proceedings of the third Nordic conference on Human-computer interaction*. ACM, 207–215.
4. Sari Kujala. 2003. User involvement: a review of the benefits and challenges. *Behaviour & information technology* 22, 1 (2003), 1–16.
5. Sari Kujala, Marjo Kauppinen, Laura Lehtola, and Tero Kojo. 2005. The role of user involvement in requirements quality and project success. In *Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on*. IEEE, 75–84.
6. Marta Kristin Larusdottir, O Haraldsdottir, and B Mikkelsen. 2009. User involvement in Icelandic software industry. In *Proceedings of the INTERACT*. Citeseer, 1–2.
7. Ramboll. 2018. SurveyXact. `https://www.surveyxact.dk/`. (2018).
8. Ken Schwaber and Jeff Sutherland. 2018. The Scrum Guide. `https://www.scrumguides.org/scrum-guide.html`. (2018). Accessed 14th of June 2018.
9. Ian Sommerville. 2010. *Software engineering*. New York: Addison-Wesley.

APPENDIX

(1) The final web questionnaire.

The final web questionnaire

1a
- Which country do you currently work in?
  - Afghanistan
  - Albania
  - Algeria
  - ...

1b
- Your occupation: (You can choose several options)
  - Developer/Programmer
  - Director
  - IT Supporter
  - IT Consultant
  - Project Coordinator
  - Project Manager
  - System Administrator
  - Other [text field]

1c
- How many employees currently work in your company?
  - 1-3
  - 4-10
  - 11-25
  - 26-50
  - 51-100
  - 101-250
  - 251-499
  - 500+
  - Don't know

[go to 2]

2a
- Which development method(s) do you use? (You can choose several options)
  - Behavior-Driven Development
  - Dynamic Systems Development Method (DSDM)
  - Extreme Programming (XP)
  - Kanban
  - Lean Software Development
  - Scrum
  - Spiral Model
  - Test-Driven Development (TDD)
  - Unified Process
  - V-Model
  - Waterfall Model
  - None
  - Other [text field]

2b
- Which stakeholders do you involve? (You can choose several options)
  - Customer
  - End User
  - Domain Expert

- Supplier
- Partner
- None [go to end]
- Other [text field]

3a
- How do you elicit initial requirements? (You can choose several options)
  - Based on government regulations
  - From focus groups with stakeholders
  - From inspiration from non-similar systems
  - From inspiration from similar systems
  - From interviews with stakeholders
  - From meetings with stakeholders
  - From requirements specification provided by stakeholders
  - From workshops with stakeholders
  - Own knowledge
  - Other [text field]

3b
- When do you involve stakeholders? (You can choose several options)
  - Requirements Analysis [show 4a & 4b]
  - Design [show 4c & 4d]
  - Implementation [show 4e & 4f]
  - Testing [show 4g & 4h]
  - Maintenance [show 4i & 4j]
  - Other [text field]

[go to relevant 4 or skip to 5]

4a
- Which methods do you use to collaborate with stakeholders during requirements analysis? (You can choose several options)
  - Brainstorming
  - Digital Prototyping
  - Focus Groups
  - Interviews
  - Meetings
  - Paper Prototyping
  - Personas
  - Questionnaires
  - Scenarios
  - Surveys
  - Use cases
  - Workshops
  - Other [text field]

4b
- Please describe how you use the methods during requirements analysis: (Optional)
  - [text field]

4c
- Which methods do you use to collaborate with stakeholders during design? (You can choose several options)
  - Brainstorming
  - Digital Prototyping
  - Focus Groups
  - Interviews

	◦ Meetings ◦ Paper Prototyping ◦ Personas ◦ Questionnaires ◦ Scenarios ◦ Surveys ◦ Use cases ◦ Workshops ◦ Other [text field]
4d	• Please describe how you use the methods during design: (Optional) ◦ [text field]
4e	• Which methods do you use to collaborate with stakeholders during implementation? (You can choose several options) ◦ Brainstorming ◦ Digital Prototyping ◦ Focus Groups ◦ Interviews ◦ Meetings ◦ Paper Prototyping ◦ Personas ◦ Questionnaires ◦ Scenarios ◦ Surveys ◦ Use cases ◦ Workshops ◦ Other [text field]
4f	• Please describe how you use the methods during implementation: (Optional) ◦ text field
4g	• Which methods do you use to collaborate with stakeholders during testing? (You can choose several options) ◦ Brainstorming ◦ Digital Prototyping ◦ Focus Groups ◦ Interviews ◦ Meetings ◦ Paper Prototyping ◦ Personas ◦ Questionnaires ◦ Scenarios ◦ Surveys ◦ Use cases ◦ Workshops ◦ Other [text field]
4h	• Please describe how you use the methods during testing: (Optional) ◦ [text field]
4i	• Which methods do you use to collaborate with stakeholders during maintenance? (You can choose several options) ◦ Brainstorming

	◦ Digital Prototyping ◦ Focus Groups ◦ Interviews ◦ Meetings ◦ Paper Prototyping ◦ Personas ◦ Questionnaires ◦ Scenarios ◦ Surveys ◦ Use cases ◦ Workshops ◦ Other [text field]
4j	• Please describe how you use the methods during maintenance: (Optional) ◦ [text field] [go to 5]

5a	• What causes problems when involving multiple, different stakeholders? (You can choose several options) ◦ Conflicting requirements among stakeholders [show 6a] ◦ Excessive engagement from stakeholders [show 6b] ◦ Lack of engagement from stakeholders [show 6c] ◦ Lack of transparency of development [show 6d] ◦ Lack of trust between developers and stakeholders [show 6e] ◦ Misunderstanding the stakeholders [show 6f] ◦ Poor prioritisation of requirements [show 6g] ◦ Other [text field] [go to relevant 6 or skip to 7]

6a	• How do you handle problems related to poor prioritisation of requirements? (Optional, but important to us) ◦ [text field]
6b	• How do you handle problems related to misunderstanding the stakeholders? (Optional, but important to us) ◦ [text field]
6c	• How do you handle problems related to conflicting requirements among stakeholders? (Optional, but important to us) ◦ [text field]
6d	• How do you handle problems related to lack of transparency of the development? (Optional, but important to us) ◦ [text field]
6e	• How do you handle problems related to lack of trust between developers and stakeholders? (Optional, but important to us) ◦ [text field]
6f	• How do you handle problems related to lack of engagement from stakeholders? (Optional, but important to us) ◦ [text field]
6g	• How do you handle problems related to excessive engagement from stakeholders? (Optional, but important to us)

- [text field]

[go to 7]

7a
- How do you validate if the software satisfies the needs of the stakeholders? (You can choose several options)
  - Acceptance testing
  - Alpha testing
  - Beta testing
  - Usability testing
  - None
  - Other [text field]

7b
- Some stakeholders do not have a representative that can speak for the entire group of that particular stakeholder, e.g. a single end user may not be able to represent a diverse group of end users.
  How do you handle this? (Optional)
  - [text field]

[end]

Involvement of Multiple Stakeholders in Scrum: A Case Study on Prescription of Medication

Menan Rajasegaram
Hobro, Denmark
mrajas13@student.aau.dk

Nicolai Vork
Aalborg, Denmark
nvork13@student.aau.dk

Søren L. Larsen
Aalborg, Denmark
slar12@student.aau.dk

ABSTRACT
Scrum has become the most popular development method in agile software development with more than half of agile adopters using it. Like other agile development methods, Scrum focuses on collaboration with the customer throughout the entire development. This is achieved with a Product Owner, who represents not only the customer, but also other stakeholders such as end users, domain experts, sponsors, and suppliers. This paper presents an empirical study of the possible involvement of multiple stakeholders and challenges that are experienced when multiple stakeholders are involved in a development process based on Scrum. We conducted a case study on prescription on medication, where we involved the following stakeholders: General practitioners, patients, pharmacists, and medical secretaries. The possible ways of involving multiple stakeholders in Scrum were through Sprint Planning and Sprint Review meetings. In both meetings, we involved several Product Owners, who represented each a given stakeholder. In Sprint Planning meetings, they prioritised and re-prioritised the user stories such that we could select user stories to implement for a given Sprint. Further, we involved the stakeholders in Sprint Review meetings. These meetings were used to gain feedback on the processed requirements in the given Sprint. We conducted separate meetings with each Product Owners and stakeholders. The challenges discovered through the case study are: Lack of time for meetings, lack of feedback on usability, difficult to set Sprint Goals, difficult to prioritise and select requirements, delay in reaching agreements, exhausted for new ideas, and difficult to involve patients.

Author Keywords
multiple stakeholders; multi-stakeholder; case study; agile software development; scrum; prescription of medication; involvement of stakeholders

INTRODUCTION

Since the introduction of the Agile Manifesto in 2001 [1], many agile development methods have emerged. Some of these are Scrum, Kanban, Scrumban, and XP. According to a survey on agile software development in 2018 [12], where 1492 individuals from the global software development community responded, Scrum has become the most popular of the agile development methods with 56 % of the responding organisations using it. Other agile development methods such as Scrumban, Kanban, and XP are much less used with 8 %, 5 %, and 1 % respectively.

Scrum, like many other agile development methods, focuses on collaborating with the customer, which is one of the values of the Agile Manifesto [1]. This is achieved in Scrum through the Product Owner, who is a single person that represents all stakeholders, including the customer [7].

In addition to the customer, there may be other stakeholders in a development project such as end users, domain experts, sponsors, and suppliers. In order for the Product Owner to represent these stakeholders, the Product Owner needs to take their viewpoints and opinions into consideration, which requires some form of involvement of these stakeholders during the development. The different stakeholders can have different viewpoints and opinions that conflict with each other, and stakeholders often do not know what they want in a system or how to express it [10, P.102]. To involve all the different stakeholders in a project can thus become a difficult task for the Scrum Team.

This paper presents an empirical study of the possible involvement of multiple stakeholders and challenges that are experienced when multiple stakeholders are involved in a development process based on Scrum.

The remaining sections of this paper are organised as follows: The next section describes existing, related work. We then describe our research method. Next, we present the findings of the study. Finally, we discuss and conclude on the contributions of this paper.

RELATED WORK

To find literature on involvement of multiple stakeholders in Scrum, we searched for papers in the databases Google Scholar and Scopus. The used keywords were ”scrum”, ”stakeholder”, ”involvement”, and ”multiple”. The search results did not yield any papers on specifically Scrum and multiple stakeholders. Therefore, we searched for papers regarding stakeholder involvement in Scrum and for papers regarding multiple stakeholders.

The handling of multiple stakeholders is a subject that has been under research for some time now. Mianabadi et al. [5]

performed a study on Group Decision Making (GDM), which relates to the handling of different opinions of multiple stakeholders. The study focuses on how to reach consensus among stakeholders with various interests, where some alternatives are equally preferred by the stakeholders. Specifically, they look into how to obtain the weights to assign to the stakeholders, when the stakeholders give their preferences on the alternatives. Thus, preferences from some stakeholders have a higher influence on the outcome of the decision making process than the preferences from others. They conclude that their proposed solution can be used to reach consensus among stakeholders through heterogeneous weights of the stakeholders.

Cajander et al. [2] conducted a study on the user perspective in Scrum projects. They performed a survey consisting of semi-structured interviews with 21 informants, who were interested in how usability can be integrated in Scrum. The study focuses on identifying the challenges of incorporating aspects of the users in projects using Scrum. They argue that Scrum does not prevent developers from involving users, but it does not explicitly support it either. They conclude that without this explicit support, the user perspective exists as informal communication, and Scrum should be changed to better integrate this perspective.

A study that also discusses usability in Scrum is a study conducted by Singh [9]. She proposed a variation of Scrum that incorporates the usability aspect during development. The proposed methodology is called U-SCRUM. It features two Product Owners: A traditional Product Owner from Scrum that focuses on the usual activities in Scrum such as backlog management, and a usability Product Owner focused on user experience and usability. Singh observed the application of U-SCRUM in the development of software products and argues that usability was improved. She concludes that, although U-SCRUM improves usability, it is best to apply it to the development of products that are not well understood, i.e. complex or novel products, since the coordination overhead of having two Product Owners could outweigh the value of the improved usability gained from using U-SCRUM.

It is clear that existing literature studies the areas of both management of multiple stakeholders and challenges of the Scrum methodology separately. However, there seems to be a need for research on how specifically Scrum deals with the involvement of multiple stakeholders during the development of a system.

THE CASE

Our case revolves around prescription of medication in the primary health care in Denmark. We are concerned with the interactions that happen between the patients, their general practitioners, medical secretaries, and pharmacists. We refer to these people as stakeholders. The four stakeholders and their interactions are shown in Figure 1.

The stakeholders interact heavily with each other in order to treat a patient the best possible way. One example of an interaction is a patient speaking with their general practitioner about the symptoms they experience. In response, the general practitioner attempts to find a cause to these symptoms in order to diagnose the patient. Another example is a pharmacist speaking to the general practitioner of a patient in an attempt to clear up a misunderstanding about information that was entered along with the prescription.

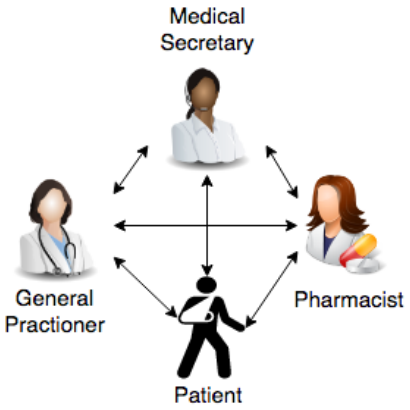


Figure 1. The interactions between the patient, medical secretary, general practitioner, and pharmacist.

To improve these interactions we focus on developing a smartphone app for patients. The goal of the app is to improve a patient's course of treatment and the efficiency of the work that general practitioners, pharmacists, and medical secretaries perform. With the app, patients are able to receive relevant information about their previous and current treatments and also perform certain actions such as requesting renewal of their prescriptions.

Existing System: Medicinkortet

An existing smartphone app for patients that relates to the case is called Medicinkortet, which is developed by Sundhedsdatastyrelsen. Patients can use the app to check active and expired prescriptions and request renewal of these. Further, patients can see how many times prescriptions can be used to dispense medication [11].

We conducted a semi-structured interview with the programme leader of Medicinkortet to know more about how they handled the involvement of multiple stakeholders during the development of Medicinkortet. The programme leader mentioned that every third Thursday, they conducted an all-day meeting with an external development team. In the meeting the developers, a project manager, a business manager, a clinical responsible, a technical responsible, and sometimes the programme leader attended. Every third month the programme leader attended a clinical forum, where general practitioners, doctors, and pharmacists attended to discuss the app. About 30 to 40 people with clinical expertise attended.

In the following we quote the programme leader, since we received the programme leader's consent to use quotes from the interview.

We asked the programme leader how they involved different types of patients and their needs, and the programme leader answered following:

We have probably not handled them [patients] as much from the start because what we agreed to do was to make a version of Medicinkortet that contained the basic part of the medicine, i.e. one can see their current medicine, one can see their prescriptions, one can request prescription renewals, and one can see information about the same on their children.

Further, the programme leader continued with following:

What we have from the patients is that we get a lot of citizen requests, that are positive, where they write that they are very happy about it, and then what they would like, and that is actually the way around we have handled it.

Citizen requests are requests through the app, Google Play reviews, App Store reviews, and emails. When we asked, if the collaboration between the patients and Sundhedsdatastyrelsen resembled an informal collaboration, the programme leader answered the following:

Yes, it has a lot to do with it being a complex area, so it is much easier for us to explain and say we can do this, and there is money for this right now, because we usually also have to make sure to have funding for it. [...] They [patients] have a lot of different wishes. [...] We also talked about making a group of 10-20 patients for testing, but it does not matter. The other [citizen requests] are actually easier [...] If you start talking to patient organisations, they all have their special interests, whereas [...] we have focused on the basic functionality of medicine, so we do not want all possible advanced solutions.

Furthermore, we asked how they tested with patients, and the programme leader answered the following:

We do not have any test groups [...] If we test with individual patients, then it's usually a mother of a developer trying it [the app] out [...] It's not because we've opted to exclude them but it's because as long as it's the basic functionality we provide, there's not really any reason to involve them.

RESEARCH METHOD

To develop the app for patients, we used a development process based on Scrum. The development is divided into two parts. The first part is requirements elicitation, and the second part is design, implementation, and evaluation. An overview of the entire development of the app is presented in Figure 2.

Requirements Elicitation

Throughout requirements elicitation we involved and collaborated with 10 patients, 5 general practitioners, and 4 pharmacists. Medical secretaries were not involved during requirements elicitation, since we did not know they were a stakeholder. We assigned three of the participants as Product Owners: One for patients, one for general practitioners, and one for pharmacists. The rest of the participants were assigned as stakeholders. The Product Owners were involved in Sprint Planning and Sprint Review meetings, whereas the stakeholders were involved in Sprint Review meetings. A full overview of when the stakeholders and Product Owners were involved in the Sprints is presented in Table 1.

According to the Scrum Guide [8], only a single person can be Product Owner in Scrum. However, since stakeholders can have different opinions that are conflicting, it is difficult for a single person to represent all the stakeholders and make informed decisions that take into account the different opinions. For this reason, we have three Product Owners instead of one, where each of them represent a stakeholder.

In our case, it was very difficult to gather several stakeholders and Product Owners at the same meeting. Many of the stakeholders and Product Owners already had jobs or other commitments, and we were not able to provide an incentive that was strong enough for them to participate, whenever we needed them to. Thus, the Sprint Planning and Sprint Review meetings with Product Owners and stakeholders were separated such that only one Product Owner or stakeholder participated in each meeting.

In order to elicit the requirements, we conducted a pre-sprint and four Sprints.

The goal of the pre-sprint was to get a better understanding of the process of prescribing medication. We conducted two semi-structured interviews: One with a pharmacist and another with a medical secretary. Further, we performed an analysis of the similar existing app Medicinkortet. Finally, we reviewed literature on prescription of medication. Based on the knowledge gained from these activities, we constructed an initial set of requirements for the app formulated as user stories.

During the first three Sprints, we held Sprint Planning meetings with the Product Owners, where they prioritised or reprioritised the requirements. Each Product Owner prioritised the requirements in six categories, namely 5, 4, 3, 2, 1, and 0, where 5 was assigned to the most important requirements to implement in the app, and 0 was assigned to requirements that should not be implemented in the app.

Based on the prioritisations from the Product Owners, some of the most important and some of the most conflictual requirements were selected for development in the Sprint. We selected the most conflictual requirements to ensure that we worked with controversy among Product Owners and stakeholders.

To find out which requirements were the most important, we added the priorities from each Product Owner for every requirement. For instance, if a requirement was categorised as 2, 4, and 5 by the Product Owners, the requirement would receive a final priority of 11. The requirements with the highest final priority were the most important ones.

To find out which requirements were the most conflictual, we added the difference between the priorities from each Product Owner for every requirement. For instance, if a requirement was categorised as 2, 4, and 5 as before, then the added difference of the priorities is $(5-4)+(5-2)+(4-2)=6$. The requirements with the highest difference between the priorities were the most conflictual ones.

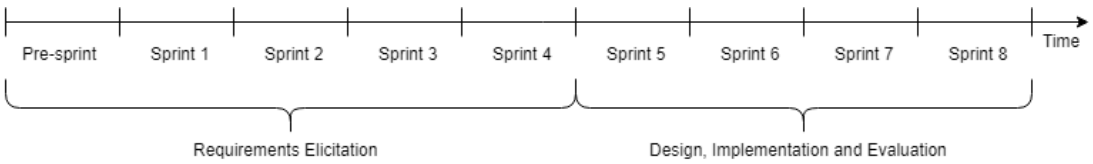


Figure 2. Timeline showing the development of the app.

Gender	Stakeholder	Role	Pre-Sprint	Sprint 1	Sprint 2	Sprint 3	Sprint 4
Female	Patient	Stakeholder		✓	✓	✓	✓
Female	Patient	Stakeholder		✓	✓	✓	
Male	Patient	Stakeholder		✓	✓	✓	
Male	Patient	Stakeholder		✓	✓	✓	
Female	Patient	Stakeholder		✓	✓	✓	
Male	Patient	Stakeholder		✓	✓		
Female	Patient	Stakeholder		✓	✓		
Female	Patient	Product Owner		✓	✓	✓	
Male	Patient	Stakeholder					✓
Female	Patient	Stakeholder					✓
Female	General practitioner	Stakeholder		✓	✓	✓	✓
Female	General practitioner	Stakeholder		✓			
Female	General practitioner	Product Owner		✓	✓	✓	
Female	General practitioner	Stakeholder					✓
Female	General practitioner	Stakeholder	✓				
Male	Pharmacist	Stakeholder	✓				
Male	Pharmacist	Stakeholder			✓		
Female	Pharmacist	Stakeholder			✓		
Female	Pharmacist	Product Owner		✓	✓	✓	

Table 1. Overview of participants during requirements elicitation.

The selected requirements were implemented as paper prototypes that each showcases how the given requirement would be implemented. We chose to use paper prototypes, since prototyping can help with the elicitation of requirements, where new ideas can emerge and lead to identification of new requirements [10, P.45].

After developing the paper prototypes, they were demonstrated to the Product Owners and stakeholders at Sprint Review meetings, where they provided feedback on the prototypes. In addition to the feedback, the stakeholders and Product Owners provided an acceptance or rejection of each requirement. The Sprint Review meetings with the stakeholders were held before the Sprint Review meetings with the Product Owners. This allowed us to pass on the feedback from the stakeholders to the Product Owners such that they could use it in their considerations for prioritisation in the following Sprint. We only passed on the stakeholders' general opinions and views on the requirements, since passing on every information would otherwise take too long.

In Sprint 4, we gathered all requirements that had been selected and accepted by the Product Owners in the previous Sprints and created a high-fidelity prototype that showcases the requirements and imitates a real smartphone app. This prototype was made with the software prototyping tool proto.io [6]. The stakeholders provided feedback on the prototype, which we used as input for the design, implementation, and evaluation of the app in addition to the feedback gathered in previous Sprints.

Design, Implementation, and Evaluation

During the design, implementation, and evaluation of the app, we involved and collaborated with 14 patients, 1 general practitioner, 1 medical secretary, and 1 pharmacist. We assigned three of the participants as Product Owners: One for general practitioners, one for medical secretaries, and one for pharmacists. The rest of the participants were assigned as stakeholders. The Product Owners were involved in Sprint Planning and Sprint Review meetings, whereas the stakeholders were only involved in Sprint Review meetings. A full overview of when the stakeholders and Product Owners were involved in the Sprints is shown in Table 2.

From the requirements elicitation we identified medical secretaries as a new stakeholder. Consequently, we assigned a medical secretary as Product Owner for medical secretaries.

As opposed to the requirements elicitation, we did not have a Product Owner for patients. We discovered that patients usually have special interests and needs in regards to the functionality of the app, and therefore a Product Owner for patients cannot represent all patients. Furthermore, we only involved patients as stakeholders, since we did not have a strong enough incentive to have general practitioners, medical secretaries, or pharmacists participate.

Gender	Stakeholder	Role	Sprint 1	Sprint 2	Sprint 3	Sprint 4
Female	General Practitioner	Product Owner	✓	✓	✓	
Male	Pharmacist	Product Owner	✓	✓	✓	
Female	Medical Secretary	Product Owner	✓	✓		
Female	Patient	Stakeholder	✓	✓	✓	
Female	Patient	Stakeholder	✓	✓	✓	
Male	Patient	Stakeholder	✓	✓	✓	
Male	Patient	Stakeholder	✓	✓	✓	
Male	Patient	Stakeholder	✓	✓	✓	
Male	Patient	Stakeholder	✓	✓	✓	
Female	Patient	Stakeholder				✓
Female	Patient	Stakeholder				✓
Female	Patient	Stakeholder				✓
Female	Patient	Stakeholder				✓
Female	Patient	Stakeholder				✓
Male	Patient	Stakeholder				✓
Male	Patient	Stakeholder				✓
Male	Patient	Stakeholder				✓

Table 2. Overview of participants during design, implementation, and evaluation.

In order to design, implement, and evaluate the app, we conducted four Sprints.

Sprint 5, 6, and 7 followed the same structure as the Sprints during requirements elicitation, but with three differences. The first difference is that in addition to prioritising requirements in Sprint Planning meetings, the Product Owners also provided a Sprint Goal to guide the design and implementation in the given Sprint. The second difference is that instead of developing paper prototypes, we developed a functional Android app using Android Studio [4], which was demonstrated to the stakeholders and Product Owners during the Sprint Review meetings. The third difference is that the requirements were only selected based on how important they are according to the Product Owners, and not how conflictual they are.

In Sprint 8, we performed a user acceptance test with patients that had not previously been involved in the project to determine whether the implementation of the requirements were adequately implemented. This test consisted of meetings with patients structured the same way as Sprint Reviews, but with a focus on whether they accept or reject the implemented requirements.

Data Collection and Analysis

During meetings with stakeholders and Product Owners we took notes about the feedback they provided. Further, we also took notes on the challenges experienced throughout the development. In total, we collected around 90 pages of notes during the development of the app. After each meeting, we reviewed the notes together and discussed them. In addition, we recorded the meeting, if we received their consent. The recordings were used to clear up any misunderstandings between us and also used to quote certain stakeholders and Product Owners. In total, we have around 22 hours of recordings.

In order to categorise the challenges encountered during the development of the app, we brainstormed on possible categories for the challenges. The result of this brainstorm was an affinity diagram [3]. We identified the following categories:

- Lack of Time for Meetings
- Lack of Feedback on Usability
- Difficult to Set Sprint Goals
- Difficult to Prioritise and Select Requirements
- Delay in Reaching Agreements
- Exhaustion of Ideas
- Difficult to Involve Patients

The categories and the related challenges are described in the following section.

RESULTS

In the following, we present the results from conducting the case study.

Identified and Processed Requirements

During requirements elicitation, we identified 26 requirements in the pre-sprint and these composed the initial requirements for the app. In the following Sprints, we identified ten new requirements in Sprint 1, two new requirements in Sprint 2, and no new requirements in Sprint 3. In the first three Sprints we worked on five requirements in each, and in Sprint 4 we worked on the 15 processed requirements from the previous three Sprints. Figure 3 presents the identified and processed requirements in Sprint 1 to 4.

During design, implementation, and evaluation, we identified no new requirements in Sprint 5, two new requirements in Sprint 6, and five new requirements in Sprint 7. In Sprint 5 we processed six requirements, and in Sprint 6 and 7 we processed five requirements in each. In Sprint 8 we worked on the 16 processed requirements from the previous three Sprints.

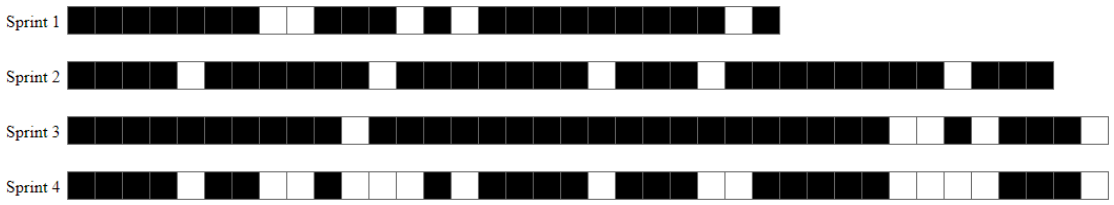


Figure 3. Overview of processed requirements during requirements elicitation. The white squares shows requirements that were proccessed, and the black squares shows requirements that were not processed.

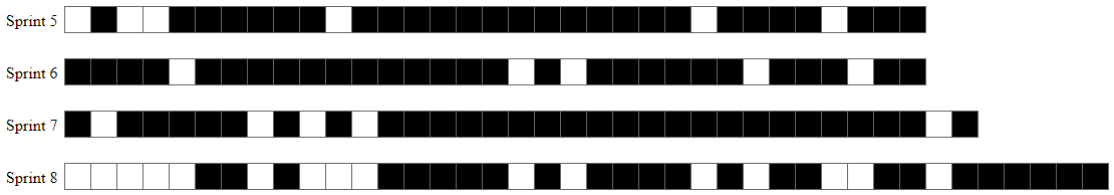


Figure 4. Overview of processed requirements during design, implementation, and evaluation. The white squares shows requirements that were processed, and the black squares shows requirements that were not processed.

Id	Gender	Age	[VGP]	[VP]	[TS]
P1	Female	19	3	1	High
P2	Male	20	2	1	High
P3	Male	22	3	0	High
P4	Female	49	10	6	Medium
P5	Male	84	14	14	Low
P6	Female	80	6	10	Medium
P7	Female	30	2	1	High
P8	Female	35	X	X	Medium

Table 3. The participating patients in the user acceptance test. [VGP]: Visit to General Practitioner (Yearly). [VP]: Visit to Pharmacy (Yearly). [TS]: Technological Skills.

Figure 4 presents the identified and processed requirements in Sprint 5 to 8.

We went from having 38 requirements at the end of Sprint 4 to 33 requirements at the beginning of Sprint 5. The reason for this is that five of the requirements identified during requirements elicitation were not related to the app for patients, but related to other systems such as the system that the general practitioner uses. Since we only focused on designing, implementing, and evaluating the app for patients, we removed the 5 unrelated requirements.

User Acceptance Test

In Sprint 8 we conducted a user acceptance test to validate, whether the app satisfies the needs of the patients. The participating patients had not participated previously in any Sprints. These are listed in Table 3.

The result of the user acceptance test was that the participated patients were satisfied with the functionalities of the app. For instance, P2 said the following about a functionality that displays a list of current and expired prescriptions: *"That's [a functionality] good, I think. [...] It gives an overview"*. Furthermore, P6 said the following about a functionality that requests renewal of a prescription: *"So you [patients] are free to sit and wait, you are number 16 in the queue. Yes, it's smart"*. Besides acceptance of the functionalities of the app, they also proposed new requirements. For instance, P5 said the following about a functionality that displays information on the selected prescription: *"The problem may be that you [patients] cannot see it [the text in the app], [...] I have had bad eyesight for some time now, so a zoom function needs to be made such that it [the text in the app] can be enlarged"*.

Lack of Time for Meetings

The Sprint Planning meetings and Sprint Review meetings, which were used to collaborate with the stakeholders and the Product Owners, were limited to around 20 minutes due to busy schedules of the stakeholders and Product Owners. This was especially the case for the pharmacists, medical secretaries, and general practitioners who had to use their working hours for the meetings. This had some implications for the requirements elicitation and the design, implementation, and evaluation of the app.

One of the implications is that only a limited amount of requirements could be processed in the Sprints, usually around 5 or 6 requirements, to allow enough time to discuss them properly at meetings.

Another implication is that some of the Product Owners experienced difficulty in having an overview and recalling the requirements, and how they prioritised them in the previous meeting. Although they were presented with the requirements and their prioritisations in front of them, the pharmacist Product Owner said the following:

"I don't really have an overview of all of them [the requirements], and I don't really recall them [the requirements]."

Although he did not have a full overview or recall all of the requirements, he attempted to re-prioritise some of them anyway. However, rarely more than a few of the requirements

were re-prioritised during all the Sprint Planning meetings if any at all.

Lack of Feedback on Usability

The feedback from stakeholders during the Sprint Review meetings for the entire development was mostly focused on whether the requirements should be implemented in the app or not. Very rarely were there any comments about the usability. During a Sprint Review meeting with a patient in Sprint 6, he noted the following about a requirement that allows patients to request renewal of their prescriptions:

"You can't renew [a prescription], if you are not allowed [by the general practitioner], and this can't be abused, so it [the requirement] is fine."

Due to the lack of feedback on usability, the design of the app did not change.

Difficult to Set Sprint Goals

When the Product Owners were asked to set a goal for the Sprint, their responses were mostly long-term goals for the entire development rather than short-term goals for Sprints. In Sprint 5, the medical secretary Product Owner set the following Sprint Goal:

"Something that provides overview, both for the patient and us [medical practice], and that is able to think by itself [...], such that when something happens in one part of the system [the app], then it [the app] also does other things in other parts of the system, so you [the medical secretary] don't have to do it manually."

Further, the Product Owners tended to repeat their goal or parts of it in later Sprints, but with different words. The following is the goal set by the medical secretary Product Owner in Sprint 7, which closely resembles the one from Sprint 5:

"You should specifically focus on intelligent solutions, that is solutions that runs by itself and don't require work from the medical practice [...], where data runs between the different systems [...], such that we [the medical practice] don't have to monitor or actively do something."

A long-term goal is helpful for the overall development of the app as a design principle, but it is too general and long-termed to tell, if we have satisfied the goal through the developed requirements at the end of the Sprint, which is the purpose of the Sprint Goal [8].

Difficult to Prioritise and Select Requirements

Since Scrum does not address how a Product Owner prioritises requirements [8], it became a difficult task to decide how this should be done, especially when we had three Product Owners. The prioritisation technique that the Product Owners used was able to prioritise the most important requirements according the Product Owners, but we had to use resources on creating the technique.

Furthermore, the Sprint Goals and priorities sometimes did not match up, which made it difficult to select requirements for a Sprint.

An example was a requirement concerning a patient being able to request renewal of their prescriptions. This requirement was specifically wanted by the pharmacist Product Owner in his Sprint Goal:

"We have had many customers who think that the cooperation between the pharmacy and the general practitioners is bad regarding, when it [the medication] is ready [for pickup], and when we [the pharmacy] should order it, so renewal of prescriptions is important."

Although the requirement was requested by the pharmacist Product Owner, the requirement did not have a high enough final priority by the Product Owners and was therefore not selected in the Sprint. However, this meant that we could not fully satisfy the Sprint Goal of the pharmacist Product Owner.

Delay in Reaching Agreements

The separated meetings made it difficult to reach agreements efficiently among stakeholders and Product Owners. In Sprint 6, a patient expressed the following new requirement during a Sprint Review meeting:

"Besides a notification, I would also like an SMS [...], because when I press it [the notification], it disappears, and I might need to remember it."

Based on this feedback, we came up with an alternative requirement that solves the problem of notifications disappearing, where patients could instead see an overview of their received notifications in the app.

In Sprint 7, the Product Owners were presented with both requirements. All Product Owners expressed that an SMS was unnecessary given patients could see an overview of their received notifications. The requirement proposed by us received a higher priority than the requirement suggested by the patient and was also selected for development.

In the Sprint Review meeting, the patient reviewed the overview of received notifications and agreed with the Product Owners. Although the patient and the Product Owners agreed in the end, it took one Sprint to reach this agreement.

Exhaustion of Ideas

During requirements elicitation fewer and fewer new requirements were identified from Sprint 1 to Sprint 4 as shown in Figure 3. To find out whether the stakeholders that had participated had been exhausted for new ideas, we conducted Sprint 4 in which some of the stakeholders had participated in previous Sprints, and some had not, as indicated in Table 1.

The feedback from the stakeholders showed that the stakeholders that had participated in previous Sprints did provide criticism, but it was along the lines of what had already been said previously. The stakeholders that had not participated previously also repeated some of the opinions that had been expressed in earlier Sprints, but also new ideas that could lead to new requirements. This indicated that the stakeholders that had participated in previous Sprints had been exhausted for new ideas, which would explain why fewer and fewer new requirements had been identified.

Difficult to Involve Patients

We found that it was particularly difficult to involve patients, since patients have specific interests, which are based on their health implications. For instance, a patient noted at the end of development that he had bad eyesight and therefore requested a zooming functionality that would enlarge the text of the app. This had not previously been requested by other patients. Thus, patients cannot be represented by a single patient or a group of patients, which is why we did not include a Product Owner for patients in the design, implementation, and evaluation of the app. Although we involved patients throughout development as stakeholders in Sprint Review meetings, we were not able to involve every aspect from them as the user acceptance test shows, where new requirements were identified.

DISCUSSION

In this Section, we discuss the choice of having several Product Owners. We also discuss the consequences of only involving patients as stakeholders. Furthermore, we relate our results to related work. We then discuss how patients were involved during the development of Medicinkortet. Finally, we discuss implications for practice.

We involved several Product Owners throughout development. An advantage of having several Product Owners, who each represent their respective stakeholders, is that they can prioritise and set Sprint Goals based on the expertise they have about the stakeholders they represent. Thus, they are able to make informed decisions, which are rooted in the expertise they have. However, as our results show, having several Product Owners also creates challenges, e.g. with selecting requirements for Sprints, where the Product Owners' prioritisations and Sprint Goals do not align. These challenges would not occur with a single Product Owner, but one risks that the single Product Owner makes uninformed decisions, because he cannot represent all diverse views from stakeholders due to lack in expertise of the stakeholders.

We only involved patients as stakeholders during the design, implementation and evaluation of the app. As a result of only involving patients, the Sprint Reviews were faster to conduct than during requirements elicitation. Although we did receive feedback from a general practitioner, a medical secretary, and a pharmacist through the use of Product Owners, they cannot express every opinion that these stakeholders have.

Related work has looked into how the user perspective is integrated in Scrum, in which it is concluded that Scrum does not have explicit support for it. From our findings, we have experienced that there was a lack of usability related feedback from the stakeholders, which could be explained by the lack of explicit support for it. Further, related work has also proposed a variation of Scrum, namely U-SCRUM, which improves the usability in Scrum projects. A similarity between U-SCRUM and our development process based on Scrum is that both have several Product Owners. Since U-SCRUM has explicit support for usability through a Product Owner specifically assigned to focus on usability, and we do not, it could explain why we lack usability feedback from stakeholders.

The programme leader of Medicinkortet noted that they probably did not handle patients much, since they agreed to only implement basic functionality. Further, they discussed having a test group of patients trying Medicinkortet out, but they argue that because they only focus on the basic functionality, there is no reason to involve patients as a test group. Instead they make Medicinkortet available to patients and let them send requests to them through citizen requests. While it is possible to involve multiple stakeholders in the development process, it is not always done. In the case of Medicinkortet, they did not actively involve patients, but rather let the patients come to them. Thus, the involvement of multiple stakeholders during development and the challenges that comes with it is a relevant area to research.

There are several implications for practitioners, who want to involve multiple stakeholders in Scrum. One implication is that practitioners should involve several Product Owners in order to represent the stakeholders that can be represented. To have a single Product Owner representing several stakeholders can be a difficult task, since they have to represent diverse views from the different stakeholders. Furthermore, practitioners should also involve stakeholders with consideration. Although stakeholder involvement can lead to the discovering of new views and opinions that had not been expressed earlier, it can also lead to exhaustion of ideas. In these cases, practitioners should consider, whether other stakeholders that had not previously been involved should be involved instead.

CONCLUSION

This paper presented an empirical study on the involvement of multiple stakeholders in Scrum, where a development process based on Scrum was used to develop an Android app for patients that improves the process of prescribing medication.

Based on the results from the case study, we conclude that it is possible to involve multiple stakeholders in a development process based on Scrum. In our development process, we involved several Product Owners in Sprint Planning and Sprint Review meetings, where each of them represented a given stakeholder. Based on the Product Owners' prioritisations, we were able to select requirements for the Sprints. Further, we involved stakeholders in Sprint Review meetings. This involvement allowed us to get feedback on the selected requirements at the end of the Sprints. Although we conducted separate meetings with each Product Owner and stakeholder, we were able to convey information between them.

We experienced a number of challenges when involving multiple stakeholders. First, there was lack of time for meetings, since they did not have extensive time for collaboration. This caused the Sprint Planning and Sprint Review meetings to be shorter, and only a limited amount of requirements could be processed in each Sprint. Second, there was a lack of feedback on usability from the stakeholders. The feedback that stakeholders provided was concerned with, whether the requirements should be implemented and why. Rarely did they address usability concerns in their feedback. Third, the Product Owners had difficulty setting proper Sprint Goals. The Sprint Goals were often long-termed and thus repeated in each Sprint. As a result, we could not tell, if we had satisfied

the the goals at the end of Sprints. Fourth, conflicting Sprint Goals and prioritisations from the Product Owners resulted in difficulty selecting requirements for development. The Sprint Goals and prioritisations sometimes did not align with each other, such that if we had to satisfy a Sprint Goal, we would have to select requirements that were not among the highest prioritised. Fifth, there was a delay in reaching agreements among stakeholders and Product Owners, which was caused by separated meetings with them. As a result of the separate meetings, they were not able to communicate directly with each other in order to reach agreements without delay. Sixth, some stakeholders became exhausted for ideas that could lead to new requirements. We experienced this during requirements elicitation, where the involvement of stakeholders that had not previously been involved were able to provide new ideas as opposed to the ones that had been involved earlier. Seventh, certain stakeholders were difficult to involve. In our case, patients could not be represented by a single person, or even a group, since they have specific interests based on the health implications they experience.

The app was both developed and evaluated by us. When evaluating the app in collaboration with the stakeholders and Product Owners, they were aware of the fact that we had developed it. This may have caused some bias that would refrain the stakeholders and Product Owners from expressing harsh criticism, since they may not want to create a negative atmosphere. Further, since we have performed a case study on prescription of medication, more case studies on the involvement of multiple stakeholders in Scrum should be performed on other cases to examine, whether similar results are experienced.

Future studies should focus on how to decide when stakeholders should be involved during development. It may be the case that a given Sprint does not benefit from involving some stakeholders, and therefore these stakeholders should not be involved in the given Sprint in order to avoid spending unnecessary resources. Further, more research should be conducted on whether certain stakeholders should be replaced with new ones and when, since the results from the requirements elicitation indicated that stakeholders had been exhausted for new ideas.

REFERENCES

1. Kent Beck, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, and others. 2001. Manifesto for agile software development. (2001).
2. Åsa Cajander, Marta Larusdottir, and Jan Gulliksen. 2013. Existing but not explicit-The user perspective in scrum projects in practice. In *IFIP Conference on Human-Computer Interaction*. Springer, 762–779.
3. Su Mi Dahlgaard-Park (Ed.). 2015. *Affinity Diagram*.
4. Google. 2018. `https://developer.android.com/studio/`. (2018). Accessed 6th of June 2018.
5. Hojat Mianabadi, Abbas Afshar, and Mahdi Zarghami. 2011. Intelligent multi-stakeholder environmental management. *Expert Systems with Applications* 38, 1 (2011), 862–866.
6. Proto.io. 2018. `https://proto.io/`. (2018). Accessed 6th of June 2018.
7. Ken Schwaber. 2004. *Agile project management with Scrum*. Microsoft press.
8. Ken Schwaber and Jeff Sutherland. 2018. The Scrum Guide. `https://www.scrumguides.org/scrum-guide.html`. (2018). Accessed 3rd of June 2018.
9. Mona Singh. 2008. U-SCRUM: An agile methodology for promoting usability. In *Agile, 2008. AGILE'08. Conference*. IEEE, 555–560.
10. Ian Sommerville. 2010. *Software engineering*. New York: Addison-Wesley.
11. Sundhed.dk. 2018. `https://www.sundhed.dk/borger/min-side/min-sundhedsjournal/medicinkortet/`. (2018). Accessed 7th of June 2018.
12. VersionOne. 2018. VersionOne 12th Annual State of Agile Report. `https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report`. (2018). Accessed 3rd of June 2018.