

**UNIVERSITY OF VAASA
SCHOOL OF TECHNOLOGY AND INNOVATION**

Roosa Salmenkylä

**RISKS OF REQUIREMENTS ENGINEERING IN DISTRIBUTED AGILE
INFORMATION SYSTEM
DEVELOPMENT**

Master's Thesis in
Information Systems

Master's Programme in
Technical Communication

VAASA 2019

TABLE OF CONTENT	page
1. INTRODUCTION	7
1.1. Purpose and objectives of the study	7
1.2. Agile development	10
1.3. Risks and challenges	11
2. INTEGRATIVE LITERATURE REVIEW	14
3. DISTRIBUTED AGILE DEVELOPMENT	20
3.1. Distributed agile software development	20
3.2. Literature on risks in distributed agile development	21
3.3. Risks and challenges of distributed agile development	29
4. AGILE REQUIREMENTS ENGINEERING	34
4.1. Requirements engineering in agile development	34
4.2. Literature on requirement risks in agile development	36
4.3. Requirements engineering challenges in agile development	43
5. SYNTHESIS	48
5.1. Common challenges	49
5.1.1. Balance of minimal documentation	50
5.1.2. Customer availability	51
5.1.3. Cost, schedule and scope estimation	52
5.1.4. Coordination challenges	53
5.2. Influencing challenges	54
5.2.1. Communication and collaboration challenges	54
5.2.2. Agile practice challenges	55
5.2.3. Challenges with tools	55
5.2.4. Complicating aspects of distributed environment	55
6. DISCUSSION	57
LIST OF REFERENCES	60
ATTACHEMENTS	67

FIGURES

Figure 1: Scope of examined risks	9
Figure 2: Risks and challenges (modified, Rames & co 2010)	13
Figure 3: Data collection	15
Figure 4: Risks & challenges in requirements engineering in agile distributed development.	49

TABLES

Table 1: Agile distributed development challenges (Alzoubi & Gill 2016)	22
Table 2: Agile distributed development challenges (Hossain & co 2009).	23
Table 3: Agile distributed development challenges (Shrivastava & Date 2010).	24
Table 4: Agile distributed development challenges (Kamaruddin & Arshad 2010).	25
Table 5: Agile distributed development challenges (Alqahtani & co 2013).	25
Table 6: Agile distributed development challenges (Thierren 2008).	27
Table 7: Agile distributed development challenges (Summers 2008).	27
Table 8: Agile distributed development challenges (Paasivaara & co 2009).	28
Table 9: Agile distributed development challenges (Kahya 2018).	28
Table 10: Agile distributed development challenges (Kajko-Mattsson & co 2010).	29
Table 11: Collected agile distributed development challenges.	30
Table 12: Agile requirements engineering challenges (Ramesh & co 2010).	38
Table 13: Agile requirements engineering challenges (Inayat & co 2015).	39
Table 14: Agile requirements engineering challenges (Alsaqaf & co 2017).	40
Table 15: Agile requirements engineering challenges (Kasauli & co 2017).	40
Table 16: Agile requirements engineering challenges (Neto & co 2017).	41
Table 17: Agile requirements engineering challenges (Pichler & co 2006).	42
Table 18: Collection of agile requirements engineering challenges	44
Table 19: Common challenges and mitigation for requirements engineering and distributed development	50

UNIVERSITY OF VAASA**School of technology and innovation****Author:**

Roosa Salmenkylä

Topic of the Master's Thesis:

Risks of requirements engineering in distributed agile information system development

Degree:

Master of Science in Economics and Business Administration

Major subject:

Master's Programme in Technical Communication

Instructor:

Tero Vartiainen

Year of Entering University:**2015****Year of Completing the Master's Thesis:** 2019**Pages: 87**

ABSTRACT

One of the critical risk elements in information system development is requirements engineering phase where the original needs are transferred to plans to be implemented. Nature of requirements engineering has been changing due to changes in business environment in past decades. Today agile development methods are typical for information system development. In addition, development is often distributed. This context has interesting aspects such as communication that is a key in agile requirements engineering and a main challenge in distributed environment.

The research area is agile information systems development and further requirements engineering and distributed environment risks. Risks of distributed agile development is mature field. Also, there are some studies about requirements engineering risks in agile development. However, there is low effort in research to examine more in detail the relation of distributed set-up and requirements engineering in agile development. The aim of this study is to identify and understand challenges of requirements engineering in distributed agile development. In addition, the study presents identified mitigation methods.

Integrative literature review was used as a research method. The method bases on existing literature that is examined to form a new framework about the topic. Utilizing this method research streams of distributed development and requirements engineering were combined. Motivation for this study originates writer's own interest due profession and instructor's study.

Synthesis was provided as a result. Common challenges for requirements engineering and distributed development in agile context were identified to be: Balance of minimal documentation, Customer availability, Cost, schedule and scope estimation and Coordination challenges. Also, mitigation methods for common challenges and impacting distributed challenges were analyzed and discussed in synthesis.

KEYWORDS: agile requirements engineering, agile distributed development

VAASAN YLIOPISTO**Tekniikan ja innovaatiojohtamisen yksikkö**

Tekijä:	Roosa Salmenkylä
Pro Gradun aihe:	Vaatimusmäärittelyn riskit hajautetussa ketterän menetelmän tietojärjestelmäkehittämisessä
Tutkinto:	Kauppätieteiden maisteri
Oppiaine:	Teknisenviestinnän maisteriohjelma
Ohjaaja:	Tero Vartiainen
Opintojen aloitusvuosi:	2015
Pro gradun valmistumisvuosi:	2019

Sivumäärä: 87

TIIVISTELMÄ

Tietojärjestelmäkehittämisen onnistumisen näkökulmasta yksi kriittisistä riskeistä on vaatimusmäärittelyvaihe. Vaatimusmäärittelyn tarkoituksena on löytää alkuperäinen tarve ja muuttaa se toteutettavaksi suunnitelmaksi. Viime vuosikymmeninä liiketoimintaympäristön muuttuessa myös vaatimusmäärittelyn toteuttaminen on muuttunut. Nykyään tietojärjestelmäkehittäminen tehdään usein ketterillä menetelmillä. Kehittäminen on yhä useammin hajautettua. Tämä yhtälö luo mielenkiintoisia asetelmia, kuten kommunikaation rooli, joka on keskeinen ketterässä kehittämisessä, mutta haaste hajautetussa ympäristössä.

Tutkimusalue on ketterä tietojärjestelmäkehittäminen ja tarkemmin vaatimusmäärittelyn ja hajautetun ympäristön riskit. Hajautetun ketterän kehittämisen riskejä on tutkittu paljon ja tutkimuksia löytyy myös vaatimusmäärittelyn riskeistä ketterässä kehittämisessä. Kuitenkaan tutkimusta ei ole tehty hajautetun asetelman vaikutuksesta vaatimusmäärittelyyn ketterässä kehittämisessä. Tämän tutkimuksen tarkoituksena on tunnistaa ja ymmärtää haasteet, joita hajautettu asetelma ketterällä menetelmällä toteutetuissa projekteissa luo vaatimusmäärittelyyn. Lisäksi tunnistetut riskien pienentämisen keinot esitellään.

Tutkimusmenetelmänä on käytetty integratiivista kirjallisuuskatsausta. Menetelmässä olemassa olevaa kirjallisuutta tutkitaan, jotta voidaan muodostaa uusi viitekehys aiheesta. Tutkimuksessa yhdistetään aiheet hajautettu kehittäminen ja vaatimusmäärittely hyödyntäen kyseistä menetelmää. Motivaatio tutkimukseen on syntynyt kirjoittajan ammatin ja ohjaajan tutkimuksen myötä.

Tutkimuksen tuloksena muodostui synteesi, joka esittää vaatimusmäärittelyn ja hajautetun kehittämisen haasteet ketterän menetelmien projekteissa. Molemmista tunnistetut haasteet ovat tasapaino minimaalisessa dokumentoinnissa, asiakkaan saatavuus, kustannusten, aikataulun ja laajuuden arviointi sekä koordinointi haasteet. Synteesissä käsitellään myös riskien pienentämisen keinoja tunnistetuille haasteille sekä hajautetun kehittämisen haasteita, jotka vaikuttavat muihin haasteisiin.

AVAINSANAT: ketterät menetelmät, vaatimusmäärittely, hajautettu kehittäminen

1. INTRODUCTION

1.1. Purpose and objectives of the study

Failure risk of information system development (ISD) projects has been considered high and it has been a topic of discussion for a long time (Keil & co 1998, Goedeke & co 2017). Risks in information system development have been studied since 1970 (Keil & co 1998). During the time business environment and needs of customer have been changing significantly. Consequently, methods of ISD have being developed and premises for risks have changed. This study concentrates understanding risks in a context that is typical for today's ISD and in a process that is critical for ISD success.

ISD methods developed massively during past decades. Methods transformed in the 1970s and the 1980s from the traditional, structured Waterfall model to agile methods which have been developing since 1990s (Tuunanen & co 2015). Main assets for agile methods are enabling changes in scope and faster releases (Holcombe 2008: 1-6). Currently there are various agile methods and plenty of existing research can be found.

International aspects and distributed context have become more common in ISD projects. Profitable, efficient and skillful workforce can be found from global markets. Distributed work environment has its own challenges including communication as the most obvious one. Research on distributed development challenges has been popular and relevant along changes in development environment. (Vallon & co 2018.)

One of the risk elements of ISD is requirements engineering (RE) phase where original needs are transferred into plans to be implemented. Misunderstanding of the requirements was ranked top 3 important risk factors in software development projects already in research published in 1998. (Keil & co 1998.) Success in this phase defines whether the end result is usable and done for right purpose. Failure in it can lead to high financial losses due to expensive defects and even unusable product. (Mc Donald 2015.) Ways of doing requirements engineering have changed a lot along the ISD methods developing.

However, the purpose of requirements engineering is still relevant as in all ISD projects the need must be transferred into solution (Mc Donald 2015). Although research is still ongoing to agree definition of agile requirements engineering.

The main idea for the research topic bases on the article of Tuunanen & co (2015) which fills a gap in research area by proposing continuous requirement risk profiling method. The research identifies risks in continuous requirements engineering. The research does not take a stand whether distributed environment or the method of the project has an effect to results. That is also the case with other researches which identify requirements engineering risks. However, that is the focus of this study as agile methods and distributed work environment are characteristic for the ISD area today.

The study examines risks of requirements engineering that are caused by distributed working environment in agile development. Topics of requirements engineering, and distributed development are combined. Overall limitation is agile ISD methods. Agile methods cover various models of working and those are evolving, but the idea behind those is the same. From risk management perspective considered steps are risk identification and control. Consequently, identified risks and challenges and ways of mitigation are being examined.

The research problem is: Which are distributed development and requirements engineering risks and challenges in agile information system development and how to mitigate those? Research questions lead from the problem are; What are the risks in distributed agile development? What are the risks of requirements engineering in agile development? Are there identified ways to control or mitigate the risks? Scope of the risks examined in this thesis can be seen in figure 1 below. Even though the limitation is definite, this kind of set-up is typical for ISD today and is faced by multiple practitioners around the world. Still no similar research is done.

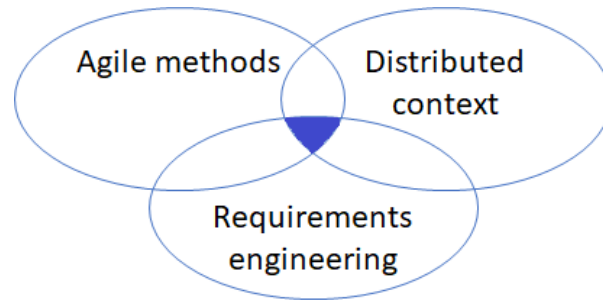


Figure 1: Scope of examined risks

Aim is to build understanding of requirements engineering challenges in agile distributed ISD projects. Existing literature is used to combine identified risks, challenges and mitigation methods of agile requirements engineering and agile distributed development. The idea is to bring up reasoned arguments and observations on topic and form a synthesis that responds to research problem. This study constitutes a concept of requirements engineering and distributed development risks in agile development.

Combining topic of requirement risks to distributed development risks grows knowledge and understanding in field of risk management. Insight is gained about topic that is faced multiple practitioners. The results can be utilized in all roles in the distributed ISD since requirements engineering touches all parties of the projects somehow. The study is also continuum to Tuunanen & co (2015) research to create deeper understanding of the requirement risks observing possible development points for created risk profiling method. In addition, research strives to indicate research gaps where more study is needed.

As stated, requirements engineering starts by identifying people's needs which relates directly to the success of the project. Therefore, requirements engineering will be difficult to replace with machines or other means. The world has become more united and global. Also, the ISD area has strong background of having international aspects. Consequently, this study combines aspects that will be relevant to ISD area now and in future.

In following paragraphs background for the topic is scanned to be able to continue to deeper research. Topics covered are Agile development and Risks and challenges. Then integrative literature review as a method and progress of the research are described in chapter 2. Later in own chapters backgrounds of Distributed development and Requirements engineering are discussed and research data is presented in chapters 3 and 4. Finally examined literature is combined into synthesis and results are discussed in chapter 5 Synthesis and chapter 6 Discussion.

1.2. Agile development

After mid-80s methods of ISD developed from traditional towards more flexible incremental and iterative approaches (Schön 2017). The driver for this was the accelerating pace of changes in business environment. In traditional methods project is conducted in sequential phases where scope and requirements are being agreed at the start of the project. Consequently, complex and stiff traditional projects do not respond to the need of fast paced environment. The solution is agile methods, which are more light weight and enable changes in requirements and scopes. (Holcombe 2008: 1-6.) In agile methods the delivery is done incrementally in short iterations. That enables changes in development according to customer need. Also, on-time delivery and customer satisfaction are potential benefits of using agile software development methods. (Schön 2017.)

Traditional and agile are umbrella terms to certain ISD methods. In this paper the term traditional method is defined as plan-based ISD method with sequential phases, such as waterfall model. Agile method is defined as incremental, co-operative, straightforward and adaptive ISD method. The describing factors for agile methods are following: development is done with small releases, communication between customer and developers is continuous, the method is easy to learn, and changes are easy to make. (Abrahamsson & co 2002.)

Most of the agile methods basis on widely known agile manifesto and principles that were brought out by a group of practitioners (Larman 2007). Agile manifesto endorses following values: individuals and interactions over processes and tools, working software over documentation, customer collaboration over contract negotiation and change over following the plan (Beck & co 2001).

There are various agile methods, new ones are being developed and in practice hybrid or modified methods are also used. Scrum is the most used agile method based on 12th Annual State of Agile report. Hybrid models are second popular (VersionOne 2018). Agile methods were developed first for small sized projects. Frameworks such as Scaled Agile Framework (SAFe) were developed to scale agile methods to be used in large organizations to synchronize multiple teams. Other scaling frameworks are Large-scale Scrum (LeSS) and Disciplined Agile Delivery (DAD). (Paasivaara 2017.) Past years SAFe have gained popularity among large organizations but not many research efforts have paid to that area yet (Putta 2018). Also, fast pace of changes is impacted ISD shifting towards constant development where systems are continuously developed. Enhancements are done constantly in streams instead of project with specific scope and requirements. (Truex 1999.)

1.3. Risks and challenges

Risk in software development means possibility to unsatisfactory outcome. Unsatisfactory outcome means different things to different parties. Risk management is the way to prevent these unsatisfactory outcomes. (Boehm 1991: 33.) Risk management is relevant to all projects and through all project phases (Avdoshin & Pesotskaya 2011). Software development risks have been studied since 1970 (Keil & co 1998: 77).

Boehm (1991: 33) divides risk management into two primary steps: risk assessment and risk control. Assessment of the risks means analyzing the effect of the potential risk and risk control includes planning the risk response and monitoring (Avdoshin & Pesotskaya

2011). This study concentrates on risk identification which is done first in risk assessment. Purpose of the identification is to find out items to compromise project success. (Boehm 1991: 33.) Management consider the most important risks are the ones that they don't have direct control (Keil & co 1998: 82). Also risk response that is part of risk control is considered as identified mitigation practices are explored. There are various ways to response to risk to mitigate it. The most effective way is to reduce potential effect or probability to occurrence in advance. (Abdul-Rahman 2012.)

Risks can be identified in different ways such as brainstorming, scenarios or other methods. Risk factor checklists are popular in risk identification due to time-saving and not requiring high-level experience. (Schmidt & co 2001). There are many studies which have identified and listed risks affecting ISD (Tuunanen & co 2015, Taylor 2012). The most well-known risk list is Boehm's top ten list of ISD risks (Keil & co 1998: 77). As discussed earlier, changing environment impacts to risks which makes older risks list possibly outdated. Taylor (2012) also states that choosing the most applicable list from various options is not simple.

Academic research results are not being utilized in practice in risk area. Risk management processes are not strictly followed. Though risk identification is usually done, and risk lists are often used to identify the risks. However, risk management recommendations are not completed only by identifying the risks. (Taylor 2012.) Taylor (2012) identifies ways of bringing research closer to practice. Key feature in exploiting research in practice is binding risk factors to project dimension. Other key features were visual presentation of research results and active participation of practitioners. As an example, Tuunanen & co (2015) combines risk factors to phases of continuous development and involves practitioners by using Delphi study method.

In this study risks and challenges are not divided as the aim is to understand possible difficulties in critical area of today's development. Challenge is something that requires special effort by its nature as per definition of Dictionary.com. Challenges vary depending on used practices. Used practices impact on what type of challenges and risks there are. If challenges are not confronted those will exacerbate risks and seriousness of the

risks. (Ramesh & co 2010.) Relations of these aspects can be seen in figure 2 in context of the study.

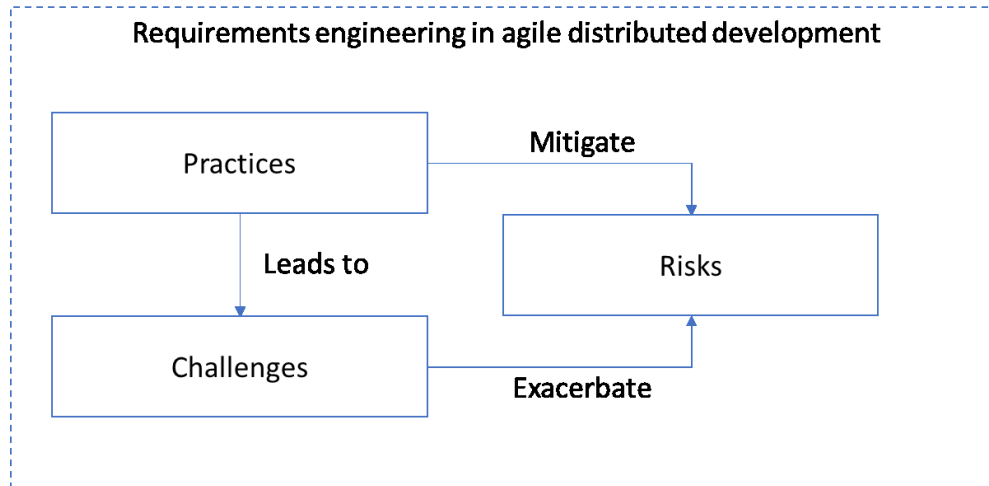


Figure 2: Risks and challenges (modified, Rames & co 2010)

2. INTEGRATIVE LITERATURE REVIEW

Integrative literature review is used as a research method. The method is based on existing literature that is examined to form a new framework or perspective about the topic (Torraco 2005). In this study new knowledge is pursued by combining developing area of agile requirements engineering and mature area of challenges in distributed IS development. As a research method integrative literature review gives possibility to review and potentially reconceptualize the topic (Torraco 2005). Because research area of the topic is partly mature and partly developing it is beneficial to combine knowledge into a framework that provides new perspective about the area. Purpose is to find out results that can be utilized widely in agile distributed ISD projects and provide new perspective and knowledge for future research. Integrative literature review as a method enables conceptualization of the area that is not yet done. As the method is literature review data is gathered from existing literature. Assertions are formed based on the data and those are led into a synthesis. Synthesis is a result of integrative literature review research and it answers to the research problem. Data collection process, analysis and forming of the synthesis are described in following paragraphs in more detail. Research problem is defined as follows:

What are distributed development and requirements engineering risks and challenges in agile information system development and how to control or mitigate those?

Research questions lead from the problem are:

1. What are the risks in distributed agile development?
2. What are the risks of requirements engineering in agile development?
3. Are there identified ways to control the risks?

The flow for research and data gathering is described in figure 3 below. Research is started with introduction part that consists of introduction to research topic and extended introduction parts in own chapters where backgrounds, meaning basic theory of the topic and the links to existing research, are explained (Hirsjärvi & co 1997: 258-259). Criteria described later are used to choose the literature for the background. The topic is split to

two main areas: distribute agile development and agile requirements engineering. The data collection continues vertically after introduction to integrative literature review that is the main research method. The criteria and research terms for the integrative literature review are defined based on the background. Finally, literature findings are combined from both areas into synthesis.

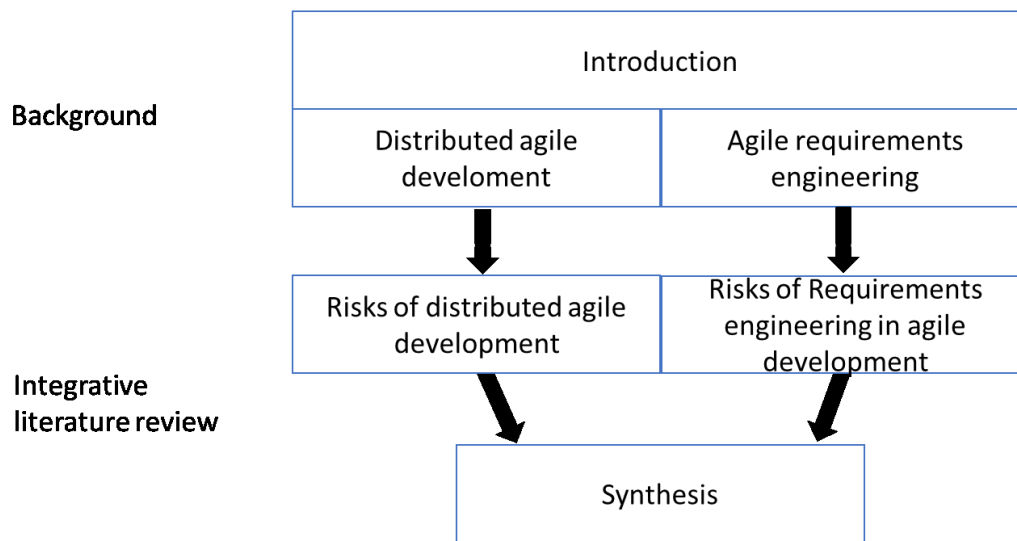


Figure 3: Data collection

Literature for both background material and integrative literature review are selected from following data bases: AIS Electronic Library (AISeL), The ACM Digital Library, IEEE explore digital library, Finna database and Google Scholar. In addition, list of references of chosen material are reviewed and used to find relevant sources. Both new and older literature is used but the concentration is on newest findings. Research done with different research methods are used but is considered in analysis.

Criteria to choose background material for introduction are derived from the topic of the study. Articles chosen with criteria are expedient and serve the matter. To be chosen article need to have at least some of the criteria in its main focus. More criteria the sources cover the better. Sources to cover all the criteria are searched. The terms used

in criteria might be different than in chosen sources since terminology in the area is varied. Used terms and their relations are explained in related paragraphs. Also, the literature is referred in related chapters. Criteria are listed here:

1. Source contains general information about distributed agile information system development.
2. Source describes how distributed agile information system development is being studied lately.
3. Source describes changes in area of distributed agile information systems development.
4. Sources contains general information about agile requirements engineering.
5. Source describes how requirements engineering is being studied lately.
6. Source describes changes in area of requirements engineering in agile information system development.

Integrative literature review is started by selecting the criteria and keywords for selecting data. Structure basis on two main parts of the research topic that are combined in the synthesis. Firstly, literature for Challenges of agile distributed ISD are found. Secondly, literature for Challenges of requirements engineering in agile ISD are found. Date base search is done for both parts. Keywords are identified based on background and categorized to be able to form the search terms. Categories and identified search terms are listed here:

Common categories for both parts

1. Information system project: information system project; ISD; software development; software project;
2. Risks: risks; challenges
3. Agile: agile; continuous;

Category part 1

1. Distributed software development: global, international, GSD, distributed

Category part 2

2. Requirements engineering: requirement

Search terms were combined with separators “OR” and “AND” into a search strings and test searches were done. Test searches were done to both Finna and AISEL. Search string did not give relevant results and it was decided to use search terms instead. Search terms were formed based on categories and test searches. “Information system project” category was not needed as other words limited the search to that area already. With word “risk” articles related to challenges were found. To limit results by agile methods both agile and continuous was used. During first data base search (AISEL) it was noticed that changing term “agile” to “continuous” did not give any new articles, that is why search with continuous was removed for next searches. During data collection search was fine-tuned and few limitations occurred. Also, option of “Matches all” of search words were tried but there was no effect. Search was first done on topic distributed agile development, category 1 and then to agile requirements engineering, category 2. Searches were done to databases mentioned before. IEEEExplore search was not working when initial searches were made. Therefore, new searches were conducted later, and search string worked well IEEE xplore database. Sorting criteria was chosen to be “relevance” and 100 first results were reviewed. Staged review was used when research was selected. Search results were pruned to the criteria 1 where title, publisher and language was reviewed. After that with criteria 2 abstract was reviewed and finally with criteria 3 the whole paper was reviewed. Criteria to choose literature was as follows:

1. Title relates to research question, published in scientific paper and language is English
2. Content based on abstract relates to research question and research is available for free
3. Critical review of the research paper: reliability and reasoning in place and research question is answered

Used literature is gathered in tables Attachment 1 and Attachment 2 that can be found at end of the paper in Attachments section. The table includes information about chosen and not chosen literature. Chosen literature is reviewed in-depth to analyze the literature.

Point of view that is used as reviewing literature is identification of challenges and risks and ways of mitigation. Meaning that research questions guides the review.

Critical analysis is done while reviewing the literature. For each selected paper main idea, perspective, research methods and limitations of the study are examined. In addition, it is explored whether studies identifying requirement risks consider also risks of distributed development already and vice versa. For validity it is considered how the risks are being identified and are the results coherent. From each selected paper list of challenges or risks and possibly mitigating methods are listed to next chapter.

In analysis risks and challenges and ways of mitigation are categorized. First similar challenges were grouped for both topics. In practice this was done by printing all the challenges on paper and then cutting pieces and grouping the physical papers. During analysis original papers were returned to clarify meaning of challenges. As similar challenges were grouped also categories formed logically from the groups. Earlier mentioned principal that results should be tied to practical project dimensions was used for naming the categories. In addition, category names from the literature was used for advice when naming categories. Also, in analysis assertions are leaded from literature. Assertions and reasoning for the assertions are added to synthesis to form a comprehensive and valid conclusion. The categorized challenges can be found relevant in chapters for both topics. Synthesis covers next chapter where topics are combined, and analysis and conclusions are presented.

Hence, literature review is written resulting in synthesis of topics that answers the research problem. Synthesis is the result of the research. Form of the synthesis is a taxonomy. A taxonomy classifies previous research (Torraco 2005). New perspective to existing results from literature is gained by classifying and comparing the challenges. Synthesis is divided into two parts. Main results are identified challenges that occurs in both topics. These common challenges are discussed based on literature and understanding of relation of the challenges and mitigation are described. The common challenges and mitigation are presented in the table. Rest of the challenges are discussed as well in order

to understand possible impacts on the development context. Finally, discussion is written to conclude aspects of the research together.

3. DISTRIBUTED AGILE DEVELOPMENT

3.1. Distributed agile software development

Amount of global agile teams has increased considerably during past years. Vallon & co (2018) informs that percent has increased from 35% in 2012 to 86% in 2016. Even though initially the most used agile methods were not developed for distributed environment. (Vallon & Co 2018.) Co-location of development and customer is core practice for many agile methods for example XP (Ramesh & co 2010:457). However, the methods have been adapted to the global environment by keeping agile values in core (Vallon & Co 2018).

In this research distributed development is defined similarly to Vallon & Co (2018) Global Software development (GSD) definition: the “development of a software artifact across more than one location.” This definition includes offshoring, outsourcing and distributed IS projects. The key is that there is no possibility to direct contact between development parties. Working together with parties that are not co-located creates challenges as there can be differences in ways of working in terms of communication, time zone, culture or other things. Used terms in this paper are GSD and distributed development.

The benefits to be achieved with GSD are savings in costs, versatile workforce, decreased time to markets and possibility to continue work around the clock (Vallon & co 2018). Costs are 35-40 % lower in developing countries compared to developed countries. These benefits make offshoring almost obligatory for companies to be able to compete in field. Markets were growing in developing countries two to three times faster than in developed countries during years 1995 to 2009. In consequence work is shifting to developing countries. (Beulen 2010.) Beulen (2010: 376-377) states that all work that does not require “a significant knowledge of business as a whole” streams offshore.

45% of respondents of 12th annual state of agile report (VersionOne 2018) use agile methods in outsourcing project and 40% are planning to increase the amount.

GSD is maturing research field and lot of research can be found (Ågerfalk & co 2009, Hossain 2009). Information System Research published special issue about flexible and distributed ISD. Issue includes research papers about topic and forms understanding about gaps on research area with Delphi study. Globalization and demand of flexibility and speed in ISD emerges new ways of working that requires more research in context of GSD. Relevant future research areas emerge as the IS landscape evolves. (Ågerfalk & co 2009). Paasivaara & co (2009) states that research combining GSD and agile methods are rare, especially case studies. However recent study of Vallon & co (2018) reviews studies from 1999 to 2016 in literature review about global software development. It concludes that agile GSD is maturing research field as well. In addition, conclusion is that contextual empirical details are needed to improve creation of generalized frameworks (Vallon & co 2018). Consequently, agile GSD as a research area is matured in recent years, meaning that basic knowledge should be found easily.

3.2. Literature on risks in distributed agile development

This chapter describes data collected about distributed agile development risks and challenges. Distributed agile development is mature research field and multiple research can be found related to the area. However, with the tight filters described in previous chapter 10 papers about challenges in agile distributed development qualified. Most of the studies are from around 2010 and all are between 2008 and 2018. Challenges were identified with different research methods. Among was two case studies, four literature reviews, two combinations of literature and empirical research and two experienced report. Areas that challenges occurred were communication, cultural differences, agile practices, time zone and premises and tools. Challenges were described in different level of accuracy. Point of view towards challenges varies: three papers are about adopting agile, two papers are about communication challenges, two papers are overall and one paper about team, geographical and Scrum method challenges. Based on research Scrum is the most

used agile method and it shows also on research area. (VersionOne 2018) Next selected papers and found challenges and mitigation strategies are described.

Alzoubi & Gill (2016) studied communication challenges in agile GSD. Research reviewed finally 22 empirical research papers where they identified 7 categories for communication challenges that are listed in table 1 below. Seven categories are Distance Differences, Customer Communication, Organizational Factors, Human Factors, Team Configuration and Project Characteristics. Motivation for research is growing interest in geographically distributed agile development and amount of papers published about communication. The systematic literature review is described precisely.

Table 1: Agile distributed development challenges (Alzoubi & Gill 2016)

Geographically distributed agile development communication challenges	Category
Time-zone differences decreasing communication opportunities	Distance Differences
Geographical differences reduce effectiveness and efficiency in communication	
Large team size	Team Configuration
Large number of teams	
Coordination	
Lack of project domain information	Project Characteristics
Lack of project architecture information between sites	
Lack of customer involvement impacting misunderstanding of requirements and developers guessing requirements	Customer Communication
Customer representative involvement	
Poor project management process weakens communication	Organizational Factors
Unsuitable communication tools	
Poor communication infrastructure	
Lack of organizational culture weakens communication and collaboration	
Language barriers	Human Factors
National culture differences (in norms, values, spoken languages and styles of communication)	
Trust in team or team members	
Personal attitudes and skills	

Hossain & co (2009) conducts systematic literature review to find out challenges occurring in use of Scrum methods in GSD projects listed in table 2 below. Motive for research is growing interest in area of agile GSD projects and boosting effective use of Scrum. Challenges are identified from twenty papers about distributed agile projects. Framework is shaped to show challenges in connection with Scrum strategies and practices. The framework is limited to project specific focus even though it is mentioned that GSD projects are usually part of bigger entities such as product portfolio. Risks and mitigation practices are listed in table below. Requirements engineering was not mentioned explicitly but most of the risks relates to communication.

Table 2: Agile distributed development challenges (Hossain & co 2009).

Risk	Risk mitigation
Inconsistent working hours	Increase amount of common working hours
	Shorten Scrum meeting length
	Divide teams to local autonomous teams
	Modified Scrum practices
Lack of team coherence	Team meeting in same location before distributed working
	Stakeholders and team members visits to distributed sites
	Informal meetings
	Trainings about Scrum
	Documentation to reduce misunderstanding
Mandatory participation	
Insufficient communication quality	Enough options for communication tools and reliable network
Lack of supportive tools	Ensure tools and support for collaboration and project management in addition to communication
Large team size	Divide into sub-teams (For example with team model such as Isolated Scrum team, Scrum of Scrum, integrated Scrum)
Shortage of collaborative environment	One room for co-located team

	Separate meeting room for distributed meetings
Large number of sites	Divide into local Scrum teams
	Limit distribution by number of sites in one Scrum team

Paper by Shrivastava & Date (2010) combines distributed and agile software development in review. Paper is discussion of challenges and practices of the topic. However, the challenges and practices are lacking connection in paper. Therefore, only challenges are noticed in this paper in table 3 below. Identification of the challenges is done based on literature and expertise of writers. Challenges are written from team point of view.

Table 3: Agile distributed development challenges (Shrivastava & Date 2010).

Agile distributed team challenges
Insufficient documentation and lack of rich conversation leads to misunderstanding
Pair programming, that is typical agile practice, is not possible
Inconsistent working hours, meaning that team members are not available at the same time
Agile practice training is not easy to organize as team is not in same place
Work distribution might lead to overspecialization divided by locations that will lead problems in components between components

Kamaruddin & Arshad (2010) have studied communication challenges in agile global development, listed in table 4 below. Starting point for study is chaotic environment for software development caused by distributed teams and agile methods where communication is crucial. Communication challenges were identified based on literature survey and feedback from a group forum. Main issues are stated to be cultural differences and insufficient face-to-face contact. No limitations are mentioned, but it is mentioned that empirical research is needed.

Table 4: Agile distributed development challenges (Kamaruddin & Arshad 2010).

Communication challenges in agile global development
Differences in project background. Different ways of working and level of knowledge on method used
Cultural differences. Such as conflicts and awareness in attitude towards negative or sensitive issues, ideology and holidays impacts in communication if not considered.
Language barriers, especially between native and not native English speaker.
Insufficient face to face contact and more asynchronous communication.
Trust issues leading to problems with collaboration and bonding
Insufficient customer involvement and concentration on processes.
Poor commitment from team leading communication to fall short and low team spirit
Inconsistent working hours leading communication gaps and scheduling problems.
Lack of requirements communication between customer and developers leads developers to conclude requirements themselves based on experience
Poor quality of communication channels
Communication costs are high and increases by quality improvements and filling the gaps.
Insufficient communication tools
Lack of versatile communication

Alqahtani & co (2013) conducted systematic literature review to identify challenges in distributed agile software development. Starting point is confrontation of agile values and distributed environment. 33 papers are reviewed that include the challenges. Challenges are categorized into 5 categories and can be found from table 5 below. Study indicates that communication is the biggest challenge followed by cultural differences. Also, it is mentioned that as a separate challenge that poor communication leads requirements misunderstanding. Research methodology is described detailed.

Table 5: Agile distributed development challenges (Alqahtani & co 2013).

Distributed agile development challenges	Category
Lack of communication and collaboration	Communication
Language barriers	

Poor communication between developers and product owners causing requirements misunderstanding	
High costs on improving communication tools and quality	
Insufficient knowledge sharing and information	
Lack of communication and collaboration due to increasing distance between agile developers	
Poor infrastructure	
Poor visibility on progress of development	
Lack of awareness due to stakeholder's cultural differences	Cultural differences
Trust issues between team members	
Poor understanding on authority for some team members	
Decreased responsibility and moral in team due to cultural differences	
Poor transparency due to cultural differences	
Productivity of developers decreases due to cultural differences	
Lack of team management	Inadequate management
Problems with cost, scope and schedule estimation	
Problems with local regulations	
Security risk due to insufficient communication	
Increased number of sites	
Lack of synchronous working time decreases time for communication in team	Time zone differences
Incoherent holiday schedules reduce common working time	
Poor agile skills	Lack of agility
Lack of development meeting	
Poor formal documentation without standards	
Too much documentation during development	
Difficulties with agile practices	
Technical issues causing neglecting agile practices and methods	

Thierren (2008) published experience report about a company adopting an agile method. Most challenges faced were on distributed teams and those are listed on table 6 below. Also, this paper highlights the importance of the communication in successful distributed team. While adopting agile methods to distributed team, team's individual features need to be considered.

Table 6: Agile distributed development challenges (Thierren 2008).

Distributed agile development challenges
In coherent working hours
Communication challenges, lack of face to face communication
Cultural differences
Trust issues within the team
Technical challenges causing problems with communication

Summers (2008) describes agile method adaption with two offshore partners in experience report. Adopted agile method was Scrum and writer worked as a Scrum Master. Point of view comes inside the project but there is no scientific evidence. Involved team members were from UK, Romania and India. The biggest challenge was stated to be cultural differences between India and other parties. Challenges were listed based on Summer's experience and can be seen in table 7 below.

Table 7: Agile distributed development challenges (Summers 2008).

Distributed agile development challenges
Cultural challenges lead to communication problems
Lack of face to face communication
Common working practices
Forming single vision of product

Paasivaara & co (2009) conducted multi case study with interviews to increase knowledge on distributed agile development. Point of view in research is how agile practices are adopted in distributed development. Also, challenges are listed for agile practices and those can be seen in table 8 below. Agile method used in case studies is Scrum. There were three cases and 19 interviews. Limitation is generalizability as the research is qualitative.

Table 8: Agile distributed development challenges (Paasivaara & co 2009).

Agile practice challenges in distributed development
Cultural differences
Incoherent holidays
Incoherent working hours limits possibility to longer meetings
Technical problems with communication channels in terms of quality
Unclarities with responsibilities with used tools (backlogs, wiki)

Kahya (2018) identifies geographical distance challenges from five distributed agile software development project in empirical research. Starting point is to examine challenges of distributed development and limitation is done to agile methods as those are widespread. Also, research suggests that agile methods can mitigate distributed development challenges. Challenges were identified from literature and in-depth interviews. Challenges are listed in table 9 below.

Table 9: Agile distributed development challenges (Kahya 2018).

Geographical distance challenges in distributed agile projects
Insufficient face to face communication
Poor team spirit
Coordination challenges
Lack of trust

Kajko-Mattsson & co (2010) examines twelve case studies from literature to gain bigger picture on challenges in distributed agile development. Paper attempts to fill gap of overall picture on challenges faced in distributed and agile development. Term problem is used instead of challenge. Challenges are categorized into six classes and can be seen in table 10 below.

Table 10: Agile distributed development challenges (Kajko-Mattsson & co 2010).

Challenge in distributed agile development	Category
Problems with responsibilities	Culture
Problems with directness and honesty	Culture
Understanding of authority	Culture
Language barriers	Culture
Incoherent holidays	Time zone
Asynchronized working hours	Time zone
Poor collaboration	Communication
Too much documentation	Communication
Effectiveness of meetings	Communication
Team spirit	Trust
Unavailability of customer	Customer collaboration
Different skill level	Training
Technical problems	Technical

3.3. Risks and challenges of distributed agile development

Challenges caused by distributed work environment are listed by identified categories in table 11 below. The challenges are combined from literature presented in previous chapter. The biggest category is communication and collaboration as expected based on background exploration. Some challenges impact on multiple other challenges. Such impacting challenges can be related to individuals, ways of working or circumstances. There are many of this kind of challenges in the list of distributed challenges especially in the category Communication and collaboration, Agile practices and Tools. There are requirements engineering related challenges in identified agile distributed challenges. Categories are discussed below the table; however entire list of distributed challenges is presented in the table 11.

Table 11: Collected agile distributed development challenges.

Challenges in distributed agile development	Category
Balance on sufficient amount of formal documentation to avoid misunderstanding (Alqahtani & co 2013, Shrivastava & Date 2010, Kajko-Mattsson & co 2010).	Requirement documentation
Unclarities with responsibilities with used tools (backlogs, wiki) (Paasivaara & co 2009).	
Customer unavailability and concentration on processes (Kajko-Mattsson & co 2010, Kamaruddin & Arshad 2010.)	Customer involvement
Lack of communication between customer and developers causing requirement misunderstanding. Developers need to conclude requirements based on experience. (Alqahtani & co 2013, Kamaruddin & Arshad 2010.)	
Communication costs increase by improving quality and tools (Alqahtani & co 2013).	Management
Cost, schedule and scope estimation (Alqahtani & co 2013).	
Poor visibility on progress of development (Alqahtani & co 2013).	
Lack of team management (Alqahtani & co 2013).	
Problems with local regulations (Alqahtani & co 2013).	
Coordination challenges (Kahya 2018).	Interaction between teams
Insufficient knowledge sharing and information (Alqahtani & co 2013).	
Increased number of sites (Alqahtani & co 2013, Hossain & co 2009).	Distribution
Work distribution might lead to overspecialization divided by locations that will lead to problems in components between components (Shrivastava & Date 2010).	
Large team size (Alzoubi & Gill 2016, Hossain & co 2009).	
Shortage of collaborative environment (Hossain & co 2009).	

Pair programming that is typical agile practice, is not possible (Shrivastava & Date 2010).	Agile practices
Technical issues causing neglecting agile practices and methods (Alqahtani & co 2013).	
Poor agile skills (Alqahtani & co 2013).	
Agile practice training is not easy to organize as team is not in same place (Shrivastava & Date).	
Lack of development meeting (Alqahtani & co 2013).	
Common working practices (Summers 2008).	
Effectiveness of meetings (Kajko-Mattsson & co 2010).	
Forming single vision on product (Summers 2008).	Team understanding
Lack of supportive and communication tools (Hossain & co 2009, Kamaruddin & Arshad 2010).	Tools
Technical problems with communication in terms of quality (Hossain & co 2009, Kamaruddin & Arshad 2010, Thierren 2008, Paasivaara & co 2009, Alqahtani & co 2013, Kajko-Mattsson & co 2010).	
Communication challenges due conflicts and awareness in attitude towards negative or sensitive issues, ideology and holidays if not considered (Kamaruddin & Arshad 2010, Thierren 2008, Paasivaara & co 2009, Summers 2008).	Communication and collaboration
Security risks due insufficient communication (Alqahtani & co 2013).	
Poor transparency (Alqahtani & co 2013).	
Productivity of developers decreases due to cultural differences (Alqahtani & co 2013).	
Poor understanding on authority for some team members (Alqahtani & co 2013, Kajko-Mattsson & co 2010).	
Language barriers (Kamaruddin & Arshad 2010, Alqahtani & co 2013, Kajko-Mattsson & co 2010).	
Lack of awareness due to stakeholder's cultural differences (Alqahtani & co 2013).	
Problems with responsibilities and moral differences (Kajko-Mattsson & co 2010, Alqahtani & co 2013).	

Problems with directness and honesty (Kajko-Mattsson & co 2010).	
Insufficient face to face communication and mostly asynchronous communication (Kamaruddin & Arshad 2010, Thierren 2008, Summers 2008).	
Inconsistent working hours due time zones and different holidays reduces common working time and possibility to longer meetings (Hossain & co 2009, Shrivastava & Date 2010, Kamaruddin & Arshad 2010, Alqahtani & co 2013, Thierren 2008, Paasivaara & co 2009, Kajko-Mattsson & co 2010).	
Trust issues impacting team collaboration (Kamaruddin & Arshad 2010, Alqahtani & co 2013, Thierren 2008, Kahya 2018).	
Lack of rich communication and collaboration (Alqahtani & co 2013, Kajko-Mattsson & co 2010, Kahya 2018, Kamaruddin & Arshad 2010, Shrivastava & Date 2010).	
Poor team spirit and commitment (Hossain & co 2009, Kamaruddin & Arshad 2010, Kahya 2018, Kajko-Mattsson & co 2010).	
Difference in project background and skill level leading to different ways of working and knowledge on method used (Kamaruddin & arshad 2010, Kajko-Mattsson & co 2010).	

There are two challenge categories that are directly linked to requirements engineering and involve distributed challenges. The categories are requirement documentation and customer involvement and those are highlighted as face to face communication that is typical for agile development becomes more difficult in distributed environment. Management will face new type of challenges in distributed development such as communication costs increases, poor visibility on progress of development or problems with local regulations. In addition, distributed environment can make typical management tasks such as estimating schedule and costs more complicated. Coordination of the work and knowledge sharing becomes more challenging in distributed development. Aspects that makes distributed environment even more complicated are listed in own category, such as large number of sites, over specialization and large team size. Agile practices that are

not developed originally for distributed development create challenges and also skill level of agile methods might vary. Team understanding category include only one challenge that is forming a single vision on product. Team category was specified as team understanding since in both topics directly team related challenges were related to understanding. However multiple other challenges impact on team and require mitigation contribution from team members. Tools and working technology become important in distributed development as they replace part of the face to face communication. Finally, communication and collaboration category include the most challenges. The category impacts on multiple other challenges that are somehow related to communication. Aspects that can be seen from Communication and collaboration challenges are cultural, individual and geographical differences and team spirit and trust. Lack of face to face communication is presented in this category as well. Here it is impact of inconsistent working hours.

4. AGILE REQUIREMENTS ENGINEERING

4.1. Requirements engineering in agile development

Requirements engineering originates from traditional development methods. Traditionally requirements engineering refers to the process of eliciting the requirements, analyzing those, writing specifications based on requirements and validating that those are correct. In addition, prioritizing of the requirements is done at the beginning and system testing is validated against the requirements in the end. (Heikkilä & co 2015.) In agile methods the steps of requirements engineering are not individual activities but those are done and repeated on each increment (Schön 2017).

The term requirements engineering is not commonly used in agile projects. Requirements engineering has a tone of inflexible and structured process that does not fit to agile world. (Ramesh & co 2010: 449-480.) However, requirements engineering activities are used in all ISD projects since system is build based on requirements (Schön 2017). Further, the mapping research of Heikkilä & co (2015) suggests that the definition of agile requirements engineering is not accurate. As the term and definition for requirements engineering emerged for traditional software development it needs to be extended and explored in context of agile methods. Many studies have tried to clarify role of requirements engineering in agile development with comparison to traditional (Ramesh 2010, Kassab & co 2018, Paetsch & co 2003). Differences between traditional and agile requirements engineering are exemplified with comparison between the methods in following chapter.

As the methods changes also the process of requirements engineering changed. Ramesh & co (2010: 449-480) describes evolution of requirements engineering process with comparison of agile and traditional development. Traditionally the starting point is that customer specifies the need. While agile project starts with assumption that initial requirements will change due to technology, customer needs or business domains. In traditional projects requirements related work is focusing in beginning and responsible

persons are named for the task. Traditional project methods separate work tasks between functions clearly. In agile projects requirements related work divide over the project timeline and assumption is that customer is available for interaction with developers. Also, changes are not significantly increasing the costs, but are part of the requirements engineering process. Changing requirements in agile projects are not documented closely. While in traditional projects purpose is that developers can easily and comprehensively understand the needs through documentation. (Ramesh & co 2010: 449-480.)

Cao & Ramesh (2008) lists seven RE practices characterized by agile development. Data was gathered from 16 companies in qualitative multi-site study. Face-to-face communication is primary custom for transferring requirements to development. User-stories or similar simple and informal documentation techniques are in use. Exception in documentation is high security applications where formal documentation is needed. However, communication towards development is still mostly oral. RE is iterative in agile development. Meaning that only high-level requirements are defined at the beginning and then features are discussed in detailed level in each development rounds. Requirements and development tasks are prioritized in each development cycles in agile RE. Planning and customer feedback is constant in agile RE. Consequently, changes in requirements are usual but major changes are rare. Prototyping is often used in agile development. It enables customer to validate requirements, process them further and helps with communication. Test-Driven Development (TDD) is agile RE practice that can be used as RE documentation that links design and production code. In TDD tests are written prior to implementation to specify how system should behave. Frequent review meetings and acceptance testing are often part of agile RE. Implemented features are being discussed in the meetings. Acceptance testing is for validation and verification. (Cao & Ramesh 2008.)

Requirements engineering have been changing but the goal is the same. Elicitation, analysis, documentation and validation are needed in agile development. However, the steps are very different in agile methods and can't be separated as such. Particularly premise for documentation is completely different in traditional and agile methods. (Paetsch & co 2003.)

Other terms are offered to describe agile requirements engineering process. Mc Donald (2015) uses term analysis and design phase to cover the requirements engineering process in agile projects. He describes definition of analysis as understanding the need, identifying solution for the need and making sure understanding of the solution is shared. Design is tool for building common understanding of wanted outcome. (Mc Donald 2015, p.xviii.) In high level this is the main idea of RE in both traditional and agile methods. Nonetheless practices are different as mentioned before.

There are lot of research about comparison between traditional and agile RE and it is commonly agreed that RE is needed in agile methods as stated before. However, there is no clear and common view on agile requirements engineering. For example, Ramesh & co (2010) and Batool & co (2013) combines traditional RE phases to agile practices but the comparison tables are not coherent. However, aim of this research is not to define agile RE. Here agile requirements engineering is examined based on agile RE practices mentioned earlier in this chapter and agile values described in chapter 1.2. Requirements engineering goals are defined as understanding the need and ensuring the understanding is shared to be able to implement the wanted outcome.

Agile requirements engineering is not a mature field but amount of research is increasing as agile methods gains popularity (Käpyaho & Kauppinen 2015). Lately agile requirements engineering studies have been concentrating in more detailed issues in addition to comparison between agile and traditional. Käpyaho & Kauppinen (2015) studies use of prototyping as an aid for RE challenges in agile development. Estimation of requirement implementation risk is being studied by Rempel & Mäder (2015).

4.2. Literature on requirement risks in agile development

Requirements risks are seen important aspect in ISD field, but with limitation to agile methods there are not that many studies about it. With selection criteria mentioned earlier 6 papers qualified. Point of view towards challenges varies in papers. Two focused

on large-scale context, two focused on comparison to traditional methods, one of the studies focused on quality requirement challenges and one case was about adapting agile methods. Papers emphasizing different focus areas increase coverage of results and understanding on different kind of project. Selected papers are described in this chapter.

Ramesh & co (2010) studies 16 software development organizations in empirical research. Research examines RE practices in agile environments and challenges in those. Semi-structured interviews, participants observations and review of documentation were methods of data collection. Data collection was done in parallel to data analysis. Analysis was done with multiple iterations and finally categories and interrelationships were identified. Six agile RE practices that resulted into seven challenges were identified from the study and discussed. Still risks used in discussion were RE risks identified for traditional methods in other research. Risks were explored from the point of view that are agile practices mitigating or increasing traditional RE risks. Main purpose for RE in traditional and agile development is the same. However, practices are different causing different kind of challenges that boosts different kind of risks. Consequently, can the same identified risks be applied directly to both approach? However, as study identifies also agile RE challenges and considers those as factors increasing risks that improves validation. Study emphasizes analysis on relationship of traditional RE risks and agile RE practices and challenges. Also, the main focus of the research is on RE practices.

As a conclusion RE practices have positive, negative and mixed impacts to traditional RE risks. Meaning that agile methods are not directly decreasing RE risks. RE risks that are increased in agile environment are overemphasizing functional requirements, insufficient requirements inspection and joining of design in requirements. Other risks were impacted also positively by agile practices. Traditional risks increased in agile environment were picked as in this research purpose is to identify risks for agile RE. With exception of risk "Issues with customer ability and concurrence among customers" that was picked as it was mentioned separately to be a risk in agile environment. Worth mentioning is also finding that the most difficult traditional risks to manage in agile development are "Issues with customer ability and concurrence among customers" and

“Neglect of non-functional requirements.” Distributed environment is mentioned in descriptions of the challenges where aspect of co-location is included. Challenges are listed in table 12 below.

Table 12: Agile requirements engineering challenges (Ramesh & co 2010).

Challenges and risks in agile RE
Issues with estimation of cost and schedule
Incompetent architecture for emerging requirements
Neglect of non-functional requirements and overemphasizing functional requirements
Customer availability and collaboration
Prioritization on a single dimension causing problems later
Insufficient requirements verification and inspection
Minimal documentation to join design and requirements
“Issues with customer ability and concurrence among customers”

Inayat & co (2015) conducted literature review on empirical research papers about RE practices in agile development. Research process of systematic literature review is described precisely in paper and finally 21 papers were reviewed. Object in the paper is to clarify concept of RE in agile methods. Research questions are: What are RE practices in agile development? What are traditional RE challenges eliminated by use of agile methods? What are agile RE challenges? Here focus is on the third research question. Identified challenges are listed in table 13. Agile RE challenges were found from: Cao & Ramesh (2008) with 16 companies in qualitative multi-site study, Ramesh & co (2010) with empirical data (interviews, participants observations, documentation review) from 16 development organizations, Pichler & co (2006) with operational case study research, Denava (2010) with 16 in-depth interviews with large organizations project workers from distributed agile projects and Ernst & co (2013) with evaluation with industry case study. Previously reviewed study by Ramesh & co (2010) is a major source for discovered challenges in the paper. Consequently, there are overlaps. Paper does not provide risk mitigation methods for agile RE risks and challenges as the main concentration is in comparison to traditional methods.

Table 13: Agile requirements engineering challenges (Inayat & co 2015).

Challenge in agile RE	Source
Minimal documentation leading to traceability issues	Cao & Ramesh (2008)
	Ramesh & co (2010)
Poor customer availability leading to rework	Ramesh & co (2010)
	Pichler & co (2006)
Unsuitable architecture causing problems in later phase and costs	Ramesh (2010)
Problems with cost and schedule estimation leading in delays	Ramesh (2010)
Neglect of non-functional requirements impacting security, usability and performance.	Ramesh (2010)
Customer inability and agreement leading to rework	Daneva & co (2013)
	Ramesh (2010)
Contractual limitations towards changes increases costs	Cao & Ramesh (2008)
Requirements change and evaluation of change leading delays	Ernst & co (2013)

Recent exploratory case study (Alsaqaf & co 2017) identifies challenges occurring in implementation of quality requirements in distributed agile projects. Data is gathered with 18 semi structured informal in-depth interviews. Study is very close to the subject of this research as context is large scale distributed agile projects. Although it is limited to quality requirements. Starting point for the research is the agile RE challenge mentioned in other researchers: neglecting of non-functional requirements. This stud focuses on quality requirement risks that can be seen in table 14 below. Research showed disagreement among practitioners about whether quality requirements should be presented as user stories or be part of the functional user stories. However, literature advocates connection between functional and quality requirements. Overall study concludes quality requirement practices to be unclear in industry. Suggestion to continue is to add view of distributed context.

Table 14: Agile requirements engineering challenges (Alsaqaf & co 2017).

Quality RE challenge in agile RE
Quality requirement find out to be not possible to implement
Interaction between teams
Insufficient verification of quality requirement
Integration testing
Loss of architectural overview
Know-how in implementation of quality requirements
Stakeholders ability to identification of quality requirements
Quality requirement visibility external and internal
Coordination between teams

Kasauli & co (2017) studies agile RE in context of large-scale development in multiple case study. Data was gathered with interviews, focus groups and workshops. There were four case companies that are large and uses agile methods. Perspective was to gain more knowledge on requirements engineering in largescale agile development where systems are often highlighted with security, regulative or other complex aspects. Two main RE challenges were found: shared understanding of value and building and maintaining of system understanding. The identified challenges are listed in table 15.

Table 15: Agile requirements engineering challenges (Kasauli & co 2017).

Challenge in agile RE	Description	Category
Shared understanding in team of customer value	In large scale development real customer and team might be far away. Sub-feature can be solution, even though it might be hard split features.	communication
Shared understanding of value in user stories	High-level user-stories could show value early. On the contrary to above mentioned challenge would benefit more targeted smaller user stories.	communication

Shared understanding on quick feedback and requirements clarification	There might be multiple stakeholders where feedback is needed. All stakeholders might not see value in quick feedback cycles.	communication
Team collaboration and synchronizing in shared requirements	Feature can be implemented by one team who synchronize the work with other teams if needed or feature can be split to multiple teams with pre-requisites to other parts.	Knowledge management
Creation and maintenance of requirements tracing	Developers might consider it too much documenting and time consuming, but same time valuable and sometimes mandatory.	Knowledge management
Gap between plan-driven and agile development	If there is not enough communication, changes are not conveyed to requirement experts. However too much communication and documenting slows done the work.	Knowledge management
Test cases and user stories not include enough information of system behavior for maintenance and other developer teams		Knowledge management
Slow and stiff tools for maintaining requirements	Tools can improve traceability and developer's motivation to documenting more precisely	Knowledge management

Neto & co (2017) focus on requirements engineering and system testing challenges in their multiple case study. Neto & co (2017) state that many of the challenges in large scale agile development issues from either requirements engineering or system testing. Paper provides first results from focus groups with topic matter experts from three companies. Companies were large-scale from telecommunication, manufacturing and automotive industries. Results were parallel to all industries. Challenges are listed in table 16. In addition, agile practices to improve alignment of requirements and testing, were provided, but not as a respond to challenges directly.

Table 16: Agile requirements engineering challenges (Neto & co 2017).

Challenge in agile RE and possible risk
--

Unclear goals leading to questionable decisions and expensive changes in later phase.
Poor cooperation in product level leading into inefficient working in terms of lead time, rework and resources between projects.
Unclear requirements leading to need of more testing efforts and failing of delivering customer value.
Insufficient quality in full test coverage and verification of requirements leading to costly changes in later phase because of not covering the whole scope in testing.
Neglecting consistent updating of requirements and tests and unalignment leads extra efforts in those if either development or testing is not up to date.
Information level consistency between documented requirements and tests can lead increased efforts and costs in testing if done poorly.
Missing trace between requirements and test cases leads to extra effort in changes
Unfit ratio between needed testing based on requirements and test resources can lead to unnecessary testing costs or poor testing quality.
Neglecting updating of requirement and test databases creates problems and false failures in testing

Pichler & co (2006) shares lessons learned from three years development project from insurance domain. Pichler & co (2006) view to requirements engineering challenges is from system development project where agile methods were introduced to customer company. Faced challenges during project are shared and also recommendations for those in table 17.

Table 17: Agile requirements engineering challenges (Pichler & co 2006).

Challenge in agile RE	Recommendations to respond to challenges
Change from traditional methods to agile	Agile development objective clarification for customer Sufficient requirement information Foster user/customer participation Foster flexibility and prioritization and that effects of changes are known
Misunderstanding in prototyping	Make sure customer understands prototyping and end product difference Foster user/customer participation
Requirements need to consider management,	Agile development objective clarification for customer Establish role of requirements engineering clearly

work flow and technology	Avoid too detailed and lengthy specification at the start of the project Foster user/customer participation Foster flexibility and prioritization and that effects of changes are known
Distributed work environment	Establish role of requirements engineering clearly Avoid too detailed and lengthy specification at the start of the project Foster user/customer participation
Different needs of stakeholders	Establish role of requirements engineering clearly Avoid too detailed and lengthy specification at the start of the project Use requirements modelling as quality assurance Foster user/customer participation Foster flexibility and prioritization and that effects of changes are known
Limited availability of customer contact	Establish role of requirements engineering clearly Foster user/customer participation
Distributed development team	Establish role of requirements engineering clearly Sufficient requirement information Use requirements modelling as quality assurance Visual models, architectures and features should come before coding at start Foster user/customer participation Select required artifacts at start and possibly use user manual as requirement documentation

4.3. Requirements engineering challenges in agile development

Many of the agile RE challenges are related to requirement documentation and stakeholder communication. There are identified challenges that are mentioned only in one study although all papers are limited to agile RE. For example, challenges caused by distributed environment and agile practices are mentioned only in one paper. Differences in result can be explained with variable research point of views in papers. RE challenges can be found in table 18 below. Categorization is done based on concrete aspects of agile development and logical grouping.

Table 18: Collection of agile requirements engineering challenges

Challenge in agile RE	Category
Balance of minimal documentation and including enough information of system behavior to user stories and test cases for traceability, maintenance and other teams (Kasauli & co 2017, Neto & co 2017, Cao & Ramesh 2008, Ramesh & co 2010, Pichler & co 2006).	Requirement documentation
Neglecting of creation and maintenance of requirement tracing causes extra effort in changes (Kasauli & co 2017, Neto & co 2017).	
Unclear requirements leading to need of more testing efforts and failing to deliver customer value (Neto & co 2017).	
Neglecting consistent updating and unalignment of requirements and tests lead extra efforts (Neto & co 2017).	
Slow and stiff tools for maintaining requirements (Kasauli & co 2017).	
Architectural overview and incorporating emerging requirements (Ramesh & co 2010, Alsaqaf & co 2017).	
Insufficient requirement verification and understanding of it leading costly changes later (Ramesh & co 2010, Neto & co 2017, Alsaqaf & co 2017, Kasauli & co 2017).	
Customer availability (Ramesh & co 2010, Pichler & co 2006).	Management
Customer inability and agreement leading to rework (Ramesh & co 2010, Daneva & co 2013, Alsaqaf & co 2017).	
Different needs and concurrence of stakeholders (Pichler & co 2006, Ramesh & co 2010).	
Contractual limitations towards changes increases costs (Cao & Ramesh 2008).	
Cost and schedule estimation (Ramesh & co 2010).	

Requirements change and its evaluation leading delays (Ernst & co 2013).	
Unfit ratio between test resources and needed testing based on requirements can lead unnecessary testing costs or poor testing quality (Neto & co 2017).	
Prioritization on a single dimension causing problems later (Ramesh & co 2010).	
Poor coordination and interaction between teams leading into ineffective working in terms of leading time, rework and resources between teams (Neto & co 2017, Kasauli & co 2017, Alsaqaf & co 2017).	Interaction between teams
Intergration testing (Alsaqaf & co 2017).	
Neglecting quality requirements can impact on security, usability and performance (Rames & co 2010).	Quality requirements
Quality requirement not possible to implement (Alsaqaf & co 2017).	
Know-how in implementation of quality requirements (Alsaqaf & co 2017).	
Quality requirement visibility external and internal (Alsaqaf & co 2017).	
Distributed work environment (Pichler & co 2006).	Distribution
Misunderstanding of prototyping (Pichler 2006).	Agile practices
Shared understanding in team on customer value (Kasauli & co 2017).	Team understanding
Unclear goals leading to questionable decisions and expensive changes in later phase (Neto & co 2017).	

Requirement documentation and stakeholder interaction are challenges but presented in agile RE practices as well. One of the defining practices in agile RE is transferring requirements from customer to development mainly via face-to-face communication instead of comprehensive documentation. Consequently, this defining agile RE practice causes also challenges. Creating and updating of documentation is time consuming.

However, some level of documentation is needed to share information and trace development and testing. Neglecting requirement tracing can cause extra effort when changes are needed. Architectural overview should be shared, and it can be challenging to incorporate emerging requirements. Extra effort is needed also if requirements are unclear or not updated or unaligned with tests. If tools for maintaining requirements are slow and stiff to use, time is wasted and motivation to maintain requirements decreases. Face-to-face communication is strongly emphasized in agile RE and it should be effective to be able to identify requirements and decrease volatility. Challenge of verification of requirements and understanding is mentioned in multiple research and can lead costly changes in later phase. Challenges for the effectiveness of communication are availability and inability of customer. Also, customer concurrence and different needs of stakeholders will cause problems.

Most of the other challenges are mentioned only in one paper, even though common topics are found. However, level of information is kept exact not to lose meaning. Contract done with fixed price can lead costly changes. Cost and schedule estimation problems can occur as there is no fixed requirements at the beginning. Also changes in later phase can cause problems with schedule. Needed test resources can be hard to estimate without fixed requirements leading either unnecessary costs or poor testing quality. If team prioritizes on one dimension of the product problems will cause later.

Poor coordination and interaction between teams was identified as challenge in three papers. Possible impacts are increased leading time, rework and inoperative resources between teams, especially when there are common requirements. Also, integration testing for quality requirements is challenging. Other challenges related to quality requirements were found as well: In later phase those might be impossible or very costly to implement, overall implementation might require specialty and visibility is needed both internally and externally. Neglecting quality requirements can cause security, usability and performance issues.

Distributed working environment was mentioned in only one paper that was lesson learned report. Also challenges related to cultural differences are not mentioned. In addition to communication and documentation related agile practices, one paper mentioned practical challenge in prototyping as customer did not understand the purpose of it. Other identified challenges related to understanding were team understanding of customer value and goals. In case of misunderstanding mistaken decisions are made and costly changes are needed in later phase.

5. SYNTHESIS

There are studies on challenges about both distributed agile development and agile requirements engineering. It is clear that challenges of both topics are faced in agile distributed information system development and challenges can impact overall project success. Here challenges of requirements engineering influenced by challenges of distributed environment are described. Also, mitigation methods are combined to challenges.

Distributed development is an aspect that brings more complexity in requirements engineering conducted with agile methods. Complexity means more challenges and need of coordination from management. There are four challenges that are directly identified to be similar in both RE and distributed development. Challenges that are common for both topic need addressing. Then there are distributed challenges that impact on development environment and ways of working and that way have influence on RE challenges. These challenges caused by distributed environment are possibly new challenges for requirements engineering or they might exacerbate the risks. The results are combined in figure 4 below. Box on the left side of the figure describes risks and challenges in requirements engineering in distributed development. Identified common challenges are listed and other requirements engineering challenges are mentioned with example. Box on the right side describes categories of influencing challenges from distributed development that may have impact on requirements engineering challenges mentioned on the left side box. Next, area of challenges is discussed from perspectives of common challenges and influencing challenges.

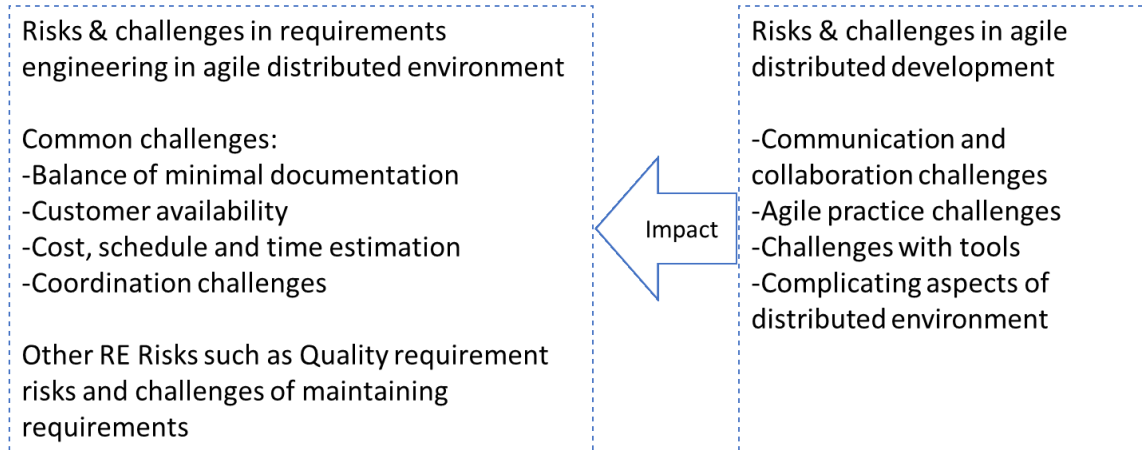


Figure 4: Risks & challenges in requirements engineering in agile distributed development.

5.1. Common challenges

Common challenges are discussed from both point of view to gain whole understanding on challenges. Following challenges are common for both requirements engineering and distributed development: Balance of minimal documentation, Customer availability, Cost, schedule and scope estimation and Coordination challenges. The challenges are presented in table 19 with relative mitigation procedures and described in text below the table. These common challenges might cause critical risk if both of the topics exacerbate the risk. On the other hand, mitigation strategies, new practices or management attention might have been grown and developed to be able to mitigate the risks. In this study the common challenges and mitigation for those are identified and relations are discussed. However, it remains for future research or practitioners to make sure the challenges are controlled.

Table 19: Common challenges and mitigation for requirements engineering and distributed development

Challenge	Description	Mitigation
Balance of minimal documentation	Documentation is time consuming and different stakeholders benefit from different type of documentation. Documentation is needed in suitable level for maintenance and distributed team members.	<ul style="list-style-type: none"> -User stories -Attach detailed descriptions -Agreed structure for documentation
Customer availability	Communication between customer and development is key to success in agile requirements engineering, but distributed environment complicates that.	<ul style="list-style-type: none"> -substitutes available for development -RE team -Agreed daily meetings with customer -Availability at the beginning of the release
Cost, schedule and scope estimation	Precise scope is not easy to estimate with agile requirements. Needed changes, resourcing and contract can cause cost and schedule problems later.	<ul style="list-style-type: none"> -Short cycle time -Continuous feedback -Constant planning
Coordination challenges	Shared requirements affecting to multiple teams are difficult to coordinate and managed which becomes more complicated in distributed environment.	<ul style="list-style-type: none"> -Cross-functional teams and meetings -Find out dependencies beforehand -Daily agile activities

5.1.1. Balance of minimal documentation

Agile methods enhance face-to-face communication, but the role of documentation grows if close collaboration towards customer and other team members decreases or disappears (Ramesh & co 2010). Level of documentation of requirements should be agreed and needs of management, developers and customer should be considered. Also, other teams and maintenance need information about system behavior and that should be taken in consideration (Kasauli & co 2017, Pichler & co 2006). User stories are not

easy to write. Customers usually benefit from larger high-level user stories while developers would benefit specific ones. Also, the more and longer user stories more maintaining is needed. (Kasauli & co 2017.) From distributed point of view lack of team collaboration leads relying on documentation as well (Kajko-Mattsson & co 2010). Distributed environment increases challenges in collaboration. Also, quality of documentation may vary without standards. (Alqahtani & co 2013.) Misunderstandings will occur without balance in documentation and collaboration (Shrivastava & Date 2010).

Both perspective state that documentation on the other hand takes time away from development while development should focus on implementation. (Ramesh & co 2010, Alqahtani & co 2013.) Inayat & co (2015) states that challenge of minimal documentation is bigger challenge in case of distributed or large-scale development. Research data refers to parallel result.

Mitigation: Requirements documented in format of user stories are communication possibility between customer and developer (Kasauli & co 2017). Documentation should be done enough to maintain communication but at the same time as little as possible. (Kajko-Mattsson & co 2010) Solutions are searched by organizations, but one possibility is to attach more detailed description, called delivery stories, to main user stories (Inayat & co 2015). Also, a structure should be agreed for documentation of requirements to make documentation coherent (Pichler & co 2010, Alqahtani & co 2013).

5.1.2. Customer availability

Analysis of requirements relies on customer interaction in agile methods. It is crucial that customer is available, and the best picture would be that developers could ask requirement related questions anytime. Otherwise requirements can't be identified and implemented correctly. (Ramesh & co 2010, Pichler & co 2010.) Requirement misunderstandings are a risk if there is not enough communication between customer and development (Alqahtani & co 2013). Forming overall vision of the product suffers without discussion opportunities. Set-up with offshore development team and onshore customer

make direct communication almost impossible. (Kajko-Mattsson & co 2010, Kamarudin & Arshad 2010.) This challenge can be concluded to be typical for distributed development and in key position for requirements analysis.

Mitigation: If developers do not have direct access to customer another person with customer access can be a substitute. The substitutes discuss about requirements and solutions with development team. (Ramesh & co 2010.) Pichler & co (2010) recommends having RE (requirements engineering) team to support with customer interaction. One method to boost customer availability is to set up daily meeting with customer and needed parties, such as developers with questions. (Ramesh & co 2010.) In addition product owner availability especially at the start of the release helps understanding requirements and product (Kajko-Mattsson & co 2010.)

5.1.3. Cost, schedule and scope estimation

Due to agile requirements engineering cost and schedule estimation for the whole scope is not easy at the beginning of the project. Agile requirements engineering means changes in scope and dynamic and iterative planning and design phases leading adjustments in estimations. Even though planning is done at the beginning some of the functionalities will be discarded and some will be added during development. (Ramesh & co 2010, Ernst & co 2013.) In addition, contract and resourcing can cause challenges in terms of cost estimation when changes to scope are done (Cao & Ramesh 2008, Neto & co 2017.) Difficulties in estimation is challenge to management. In addition to requirements engineering challenges at the beginning of the project, progress of the development is not easy to follow in distributed environment. In distributed development also, communication related costs can raise high if improvements to tools or infrastructure are need or visits to sites that are far away are required. (Alqahtani 2013.) These estimation challenges were discussed only in one study of the distributed data. Still the aspect is worth to recognize.

Mitigation: Short cycle time and continuous feedback makes easier to create estimates for one cycle. It should be considered that estimation in agile development can be done

for one cycle but for whole scope it is challenging. (Ramesh & co 2010.) Constant planning is typical for agile development and in addition to continuous feedback, those minimize probability of need for large changes (Cao & Ramesh 2008).

5.1.4. Coordination challenges

In agile RE coordination challenges are mentioned to exist between teams, in product level (Neto & co 2017). Especially shared requirements and backlog require coordination (Kasauli & co 2017, Alsaqaf & co 2017). Poor coordination between teams leads to ineffectiveness and poor lead time, rework and poor resources optimization. (Neto & co 2017) In addition coordination between teams is especially important in terms of quality requirements as those often affect the whole system (Alsaqaf & co 2017). Distributed environment increases coordination challenges inside teams due to physical and possible temporal distance of team members. Consequently, challenges between teams in requirements engineering might become a problem also inside a team in case a team is distributed. Also, knowledge and information sharing may be inadequate. (Kahya 2018, Alqahtani & co 2013) Lack of coordination activities from management will create challenges. Distributed aspects such as regulations of other countries or multiple sites create more coordination work for management. (Alqahtani & co 2013.)

Mitigation: Cross-functional teams improve coordination and collaboration in product level. Also, coordination can be improved by involving a variety of roles and expertise to expand knowledge sharing and interaction (Neto & co 2017). If the whole feature is implemented by one team coordination is needed only for dependencies. However, then dependencies must be known in advance. Also, quality requirements must be considered in this case. (Kasauli & co 2017) Further in distributed development coordination is needed between distributed parties as well. Daily agile activities, such as daily Scrum, improve coordination between team members. Another meeting involving management improves collaboration towards other stakeholders. (Kahya 2018.) This requires knowledge and skills of agile practices that is also a challenge in distributed development.

5.2. Influencing challenges

This chapter discusses distributed challenges that are not widely identified as requirements engineering challenges based on results. The relationship of these challenges might be complicated. As distributed challenges are raising from the environment and circumstances those might have wider impact to more context focused requirements engineering challenges. In addition, those might become or create new challenges. Identified distributed challenge categories with wider impact are communication and collaboration challenges, agile practice challenges, challenges with tools and challenges making set-up complicated. Challenges are presented in categories because those describe a high-level impact of the grouped challenges. The categories are listed earlier in figure 4, related challenges are listed earlier in tables 11 and 18 by categories and discussed in next chapters.

5.2.1. Communication and collaboration challenges

There are multiple communication and collaboration related challenges in distributed development. Identified communication related challenges in distributed agile development are related to mainly cultural and geographical differences and team spirit or trust in teams. It is mentioned in different research that agile methods that encourages communication could reduce communication problems in distributed development. (Vallon & co 2018). Still many researches identify communication challenges in distributed agile development as seen from the results. What does this mean for requirements engineering activities as communication is key element in agile requirements engineering? Does this create new challenges or exacerbate existing challenges related to communication? Communication and collaboration related challenges are identified in more detail in results from distributed development than from requirements engineering. In distributed development there are also challenges related to features of individuals while requirements engineering results concentrate more in requirements engineering activities and not individuals carrying out the activities. However, identified requirements engineering challenges involve communication and collaboration and impact of individuals' actions. Those are also enhanced in Agile methods as mentioned in chapter 1.2

about agile development. Therefore, agile requirements engineering can be deduced to be impacted with the distributed challenges of communication and collaboration category. Still width, severity or intensity of impact is not possible to specify with these results. Shared understanding of customer value or interaction between teams are examples of requirements engineering challenges impacted with communication and collaboration challenges. Agile practices boosting communication could be concluded to be in important role in mitigating these challenges.

5.2.2. Agile practice challenges

As mentioned in introduction, practices can prevent challenge to become a risk and mitigate risks. This study is limited to use of agile methods which means it is likely that agile practice challenges impacts on requirements engineering challenges. Agile practices challenges are related to knowhow of agile practices and difficulties to carry out agile practices due distributed environment or tools. These challenges might exacerbate other challenges in agile requirements engineering. In addition, there are existing practices and entire methods for agile distributed development that could be examined in relation to the results of this study. This way understanding on specific method could be build deeper and best solutions could be discovered.

5.2.3. Challenges with tools

Identified challenges with tools are mostly practical problems. Challenges with tools emerges if there are technical problems with those or there are not sufficient tools to support the work. These can be usually fixed but it requires money. Impact might be wide in case the challenges slow down the work. Also this challenge might exacerbate communication related challenges.

5.2.4. Complicating aspects of distributed environment

There are few aspects that are known to make distributed environment more complicated. The larger number of sites there are the more coordination is needed. Challenge

is also to make sure sites do not overspecialize in own parts of the project. That causes parts in between being missed. Finally, large team size makes working in distributed environment more complicated.

6. DISCUSSION

In this integrative literature review about challenges of requirement engineering and distributed environment existing literature is logically combined to create a view on the topic that has not been studied before. Main results are identified common challenges for agile requirements engineering and distributed development with limitation to agile development. Findings shows critical points to consider in agile distributed requirements engineering and provides mitigation activities identified in literature. The common challenges are Balance of minimal documentation, Customer availability, Cost, schedule and scope estimation and Coordination challenges. The common challenges with description and mitigation methods can be found in table 19. In addition, exacerbating aspects and possible new challenges for requirements engineering due distributed environment are presented in chapter 5.2 Influencing challenges. The synthesis, that is a result of this study, builds understanding on complex topic combining requirements engineering, distributed development and agile methods. Also, new point of views and open questions are found providing interesting grounds for future research. Success in the area of this study is important for today's and future information system projects. In practice the results can be utilized in risk management to decrease risk of failure. Research with similar limitation is not found in current literature. However, this study can be seen as continuum to existing literature as existing literature is used here to gain understanding on more detailed topic.

There are various development set-ups in information system development. Methods from various options are applied different ways in different situations. Premises for development varies such ways as size and budget of the project, level of experience and fixed processes and ways of working. Challenges and risks are different in different kind of information system development. This study describes possible challenges and critical aspects to be faced in requirements engineering in distributed agile development. Ignoring challenges can lead to decreasing advantages of agile development or realized risks (Alqahtani 2013). Agile RE requires careful planning to success (Ramesh & co 2010). Understanding on this topic helps management in similar situation towards successful delivery. Management should utilize this knowledge from the beginning of the

project and improve risks management. Also, participants of all roles from customer to developers who are participating in agile distributed development are recommended to raise proficiency with knowledge and understanding provided in the research. Practitioners working with method development could also benefit from results.

For researchers the study opens new opportunities to go deeper in successful distributed development. One interesting stream would be research on how different methods and practices deal with the identified challenges. Hussain (2018) raises lack of distributed requirements engineering related research in terms of identified challenges as gap in his analysis in study about requirements engineering in Scaled Agile Framework. This kind of research would provide guidelines for further development of methods. Also, Ramesh & co (2010) have studied relations of practices, risks and challenges in requirement context. It would be beneficial to add distributed point of view to that type of study.

Biases tried to be avoided by concentrating strictly on research questions. Bias could have formed if distributed development challenges would have directly concluded to be challenges in requirements engineering. However, as integrative literature review gives an opportunity to analyze and creatively combine data, it was possible to analyze each challenge and challenge categories based on literature from both topics and present conclusions. These conclusions should be further studied or applied in practice depending on situation. Anyway, each development setup has own characteristics that affects the situation. Seriousness or prioritization for risks and challenges is not concluded since risk for biases could be high as the research is qualitative and not concentrating on those aspects.

The study succeeds to answer to research problem and bring understanding on the new topic as aimed. An aspect that could have been studied with different kind of methods is ways of mitigation. As mentioned in introduction, practices can lead mitigation of the risk and challenges meaning that correlation of specific practices and mitigation could be investigated. However, that would have expanded the study over the resources. It is stated that main focus is on risks and challenges. Literature was limited

strictly based on research problem and questions. Practitioners are not involved in the research except indirectly as used literature included empirical research. The study could be continued with empirical research.

It is possible that all the relevant articles from the research area were not found when literature was searched. Multiple data bases were searched but there were some papers with no free access and human errors are possible. Writer works with requirements engineering that gives professional view to interpretation. Writers own competence was utilized in understanding the challenges and concluding the results. Overall validity and reliability are well accomplished considering resources. Focus of the study is on challenges meaning that all of the known mitigation methods are not presumably captured. Research was conducted with good research ethics. Still, there is possibility for human errors, especially as there was only one researcher. Overall, the research contributes by creating a new perspective to research area and useful practical advices to practitioners.

LIST OF REFERENCES

- Abdul-Rahman, H., F. Mohd-Rahim & W. Chen (2012) *Reducing failures in software development projects: effectiveness of risk mitigation strategies* [online] Journal of Risk Research. Apr2012, Vol. 15 Issue 4, p417-433. 17p. [cited 1st Aug. 2018] DOI 10.1080/13669877.2011.634520
- Abrahamsson P., O. Salo, J. Ronakinen & J. Warsta (2002) *Agile software Development Methods: Review and Analysis* [online, authors version] VTT publication 478, Espoo, Finland, 107p [cited 5th of Jan. 2019] Available from World Wide Web: URL:<https://arxiv.org/ftp/arxiv/papers/1709/1709.08439.pdf>
- Alqahtani, A., J. Moore, D. Harrison & B. Wood (2013) *The Challenges of applying distributed agile software development: A systematic review* [online] International journal of advances in Engineering & Technology, Jan 2013, Vol. 5, issue 2, pp. 23-36. [cited 5th of Jan. 2019] ISSN: 2231-1963
- Alsaqaf, W., M. Daneva & R. Wieringa (2017) Agile quality requirements engineering challenges: first results from a case study [online] 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement [cited 7th Mar. 2019] DOI: 10.1109/ESEM.2017.61
- Alzoubi, Y., A. Gill & A. Al-Ani. (2016) *Empirical studies of geographically distributed agile development communication challenges: A systematic review.* [online] Information and Management 1 January 2016, Vol.53(1), pp.22-37. [cited 23rd Aug. 2018] <https://doi.org/10.1016/j.im.2015.08.003>
- Avdoshin, S. & E. Pesotskaya (2011) *Software risk management.* [online] 2011 7th Central and Eastern European Software Engineering Conference (CEE-SECR). [cited 7th Mar. 2019] DOI: 10.1109/CEE-SECR.2011.6188471

- Batool A., Y. Motla, B. Hamid, S. Asghar, M. Riaz, M. Mukhtar & M. Ahmed (2013) *Comparative study of traditional requirement engineering and Agile requirement engineering* [online] 2013 15th International Conference on Advanced Communications Technology (ICACT) [cited 4 Mar. 2019] ISBN 978-89-968650-1-8
- Beck K., M. Beedle, A. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. Martin, S. Mellor, K. Schwaber, J. Sutherland, D. Thomas (2001) *Manifesto for agile development* [online] [cited 4 Mar. 2019] Available from World Wide Web: <URL: <https://agilemanifesto.org/>>
- Beulen E. (2010) *Long-held perceptions of the consequences of IT offshoring will become a reality: fewer IS jobs in developed countries* [online] Journal of Information Technology, suppl. Special Issue: 25th Anniversary Edition; London Vol. 25, Iss. 4, (Dec 2010): 376-377. [cited 1st Aug. 2018] DOI:10.1057/jit.2010.36
- Boehm B. (1991) *Software Risk Management: Principles and Practices* [online] IEEE Software Volume: 8, Issue: 1, Jan. 1991 p 32-41. DOI 10.1109/52.62930 [cited 1st Aug. 2018] DOI: 10.1109/52.62930
- Cao, L., & B. Ramesh (2008) *Agile Requirements Engineering Practices: An Empirical Study* [online] IEEE Software Volume: 25 , Issue: 1 , Jan.-Feb. 2008. [cited 9 Aug. 2018] DOI: 10.1109/MS.2008.1
- Goedeke, J., O. Mueller & O. Pankraz (2017) *Uncovering the Causes of Information Systems Project Failure* [online] Twenty-third Americas Conference on Information Systems, Boston [Cited 10th Dec. 2018] Available from World Wide Web: <URL: <https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1246&context=amcis2017>>

- Heikkilä V., D. Damian, C. Lassenius & M. Paasivaara (2015) *A Mapping Study on Requirements Engineering in Agile Software Development* [online] 41st Euromicro Conference on Software Engineering and Advanced Applications 2015 [cited 9 Aug. 2018] DOI: 10.1109/SEAA.2015.70
- Hirsjärvi, S., P. Remes & P. Sajavaara, P (1997). *Tutki ja kirjoita*. 22nd edition. Helsinki: Tammi. p. 258-259.
- Holcombe, M. (2008) *Running an agile software development project*. John Wiley & Sons, Incorporated. DOI: 10.1002/9780470385883
- Hossain, E., M. Babar, H. Paik, & J. Verner (2009) *Risk Identification and Mitigation Processes for Using Scrum in Global Software Development: A Conceptual Framework* [online] 2009 16th Asia-Pacific Software Engineering Conference [cited 8 Mar 2019] DOI 10.1109/APSEC.2009.56
- Hussain, N. (2018) *Requirements Engineering for Globally Distributed Teams using Scaled Agile Framework*. [online] Aalto University School of Science, Master's thesis. [cited 18th May 2019] https://aaltodoc.aalto.fi/bitstream/handle/123456789/32500/master_Hussain_Nazia_2018.pdf;jsessionid=5417BE1A50F1B074006892FFECF61FAF?sequence=2
- Inayat, I., S. Salim, S. Marczak, M. Daneva (2015) *A systematic literature review on agile requirements engineering practices and challenges* [online] Computers in Human Behavior 1 October 2015, Vol.51, Part B, pp.915-929. [cited 7 Mar 2019] <https://doi.org/10.1016/j.chb.2014.10.046>
- Kahya M., C. Seneler (2018) *Geographical Distance Challenges in Distributed Agile Software Development: Case Study of a Global Company* [online] 2018 3rd International Conference on Computer Science and Engineering (UBMK) [cited 6 Mar 2019] DOI 10.1109/UBMK.2018.8566591

- Kajko-Mattsson, M., Azizyan G., Magarian M. (2010) *Classes of Distributed Agile Development Problems* [online] 2010 Agile Conference [cited 6 Mar 2019] DOI 10.1109/AGILE.2010.14
- Kamaruddin, N., N., Arshad, A. Mohamed (2012) *Chaos issues on communication in agile global software development* [online] 2012 IEEE Business, Engineering & Industrial Applications Colloquium (BEIAC). [cited 6 Mar 2019] DOI: 10.1109/BEIAC.2012.6226091
- Kasauli, R., G. Liebel, E. Knauss, S. Gopakumar & B. Kanagwa (2017) *Requirements Engineering Challenges in Large-Scale Agile System Development* [online]. IEEE 25th International Requirements Engineering Conference [cited 7 Mar. 2018] DOI 10.1109/RE.2017.60
- Keil, M., P. Cule, K. Lyytinen & R. Schmidt (1998). *A framework for identifying software projects risks* [online]. Communications of the ACM 1998 [cited 8 Jun. 2018] DOI: 10.1145/287831.287843
- Larman G. (2007) *Agile and iterative development*. 9th edition. Addison-Wesley Professional. ISBN-0131111558.
- Mc Donald, K. (2015) *Beyond Requirements: Analysis with an Agile Mindset (Agile Software Development Series)* 1st Edition. Addison-Wesley Professional; 1 edition (September 21, 2015) ISBN-10: 0321834550
- Neto, F., J. Horkoff, E. Knauss, R. Kasauli & G. Liebel (2017) *Challenges of Aligning Requirements Engineering and System Testing in Large-Scale Agile: A Multiple Case Study* [online]. 2017 IEEE 25th International Requirements Engineering Conference Workshops [cited 8 Mar. 2019] DOI 10.1109/REW.2017.33

- Paasivaara, M. (2017) *Adopting SAFe to Scale Agile in a Globally Distributed Organization* [online]. 2017 IEEE 12th International Conference on Global Software Engineering (ICGSE). [cited 24 Feb. 2019]. DOI: 10.1109/ICGSE.2017.15
- Paasivaara, M, S. Durasiewicz & C. Lassenius (2009) *Using Scrum in Distributed Agile Development: A Multiple Case Study* [online]. 2009 Fourth IEEE International Conference on Global Software Engineering. [cited 24 Feb. 2019] DOI: 10.1109/ICGSE.2009.27
- Pichler M., H. Rumetshofer & W. Wahler (2006) *Agile Requirements Engineering for a Social Insurance for Occupational Risks Organization: A Case Study* [online]. 14th IEEE International Requirements Engineering Conference (RE'06) [cited 8 Mar. 2019] DOI 10.1109/RE.2006.8
- Putta, A., M. Paasivaara & C. Lassenius (2018) *Benefits and challenges of adopting the Scaled Agile Framework (SAFe): Preliminary results from a multivocal literature review* [online] Abstract. PROFES 2018: Product-Focused Software Process Improvement pp 334-351. [cited 12 Mar. 2019] DOI https://doi.org/10.1007/978-3-030-03673-7_24
- Ramesh, B., L. Cao, & R. Baskerville (2010) *Agile requirements engineering practices and challenges: an empirical study* [online]. Information Systems Journal, 2010 [cited 26 Jun. 2018]. DOI: 10.1111/j.1365-2575.2007.00259.x
- Schmidt R., K. Lyytikäinen, M. Keil & P. Cule (2001) *Identifying software project risks: An international Delphi study* [online] Journal of Management Information Systems: JMIS; Armonk Vol. 17, Iss. 4, (Spring 2001): 5-36 DOI 10.1080/07421222.2001.11045662
- Schön, E., J. Thomaschewski & M. Escalona (2017). *Agile Requirements Engineering: A systematic literature review* [online]. Computer standards & interfaces 2017 [cited 26 Jul. 2018]. DOI: <https://doi.org/10.1016/j.csi.2016.08.011>

- Shrivastava, S. & U. Rathod (2015) *Categorization of risk factors for distributed agile projects* [online] *Information and Software Technology* 58 (2015) 373–387 [cited 8 Mar. 2019]. DOI: <http://dx.doi.org/10.1016/j.infsof.2014.07.007>
- Shrivastava, S. & H. Date (2010) *Distributed Agile Software Development: A Review* [online] *Journal of Computer Science and Engineering*, Volume 1, Issue 1, p10-17, May 2010 [cited 8 Mar. 2019] Available from World Wide Web: <http://sites.google.com/site/jcseuk/volume-1-issue-1-may-2010>
- Summers, M. (2008) *Insights into an Agile Adventure with Offshore Partners*. [online] Agile 2008 Conference. [cited 8 Mar. 2019] DOI: 10.1109/Agile.2008.37
- Taylor, H. (2012) *Information technology project risk management: bridging the gap between research and practice*. *Journal of Information Technology* 27(1):17-34. March 2012 [cited 8 Mar. 2019]. DOI: 10.1057/jit.2011.29
- Therrien, E. (2008) *Overcoming the Challenges of Building a Distributed Agile Organization* [online] Agile 2008 Conference IEEE 10.1109/Agile.2008.9 [cited 6 Mar 2019]
- Torraco, R. (2005). *Writing Integrative Literature Reviews: Guidelines and Examples* [online]. Sage Publications: Human Resource Development Review 2005 [cited 8 Jun. 2018]
- Truex, D., R. Baskerville & H. Klein (1999) *Communications of the ACM* [online] August 1999/Vol. 42, No. 8 [cited 8 Jun. 2018] DOI 10.1145/310930.310984
- Tuunanen, T., T. Vartianen, M. Ebrahim & M. Lang (2015) *Continuous Requirement Risk Profiling in Information Systems Development* [online]. 2015 48th Hawaii International Conference on System Sciences. [cited 9 Aug. 2018] DOI: 10.1109/HICSS.2015.483

Vallon, R., B. Jose da Silva Estacio & R. Prikladnicki (2018) *Systematic literature review on agile practices in global software development* [online] Information and Software Technology 2018 [cited 12 Jul. 2018] DOI: <https://doi.org/10.1016/j.infsof.2017.12.004>

VersionOne (2018) *12th annual state of agile report* [online] CollabNet VersionOne May 12 2018 [cited 24 Feb. 2019] Available from World Wide Web: <https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report>

Ågerfalk, P., J. Fitzgerald, B. Slaughter, A. Sandra. (2009) *Introduction to the Special Issue: Flexible and Distributed Information Systems Development: State of the Art and Research Challenges*. Information Systems Research; Linthicum Vol. 20, Iss. 3, (Sep 2009): 317-328. ISSN: 10477047.

ATTACHEMENTS
ATTACHEMENT 1.

Data base	Search word/string	Results	Reviewed	Date of search	Double	Criteria 2 fail	Reason 2	Criteria 3 fail	Reason 3	All criteria passed	Writer	Title	Published
AIS Electronic Library (AISeL)	risk agile global	16	16	23.8.2018				3	SLR papers not limited to agile		Alzoubi, Yehia Ibrahim and Gill, Asif Qumer	Agile global software development communication challenges: A systematic review	18th Pacific Asia Conference on Information Systems, PACIS 2014
						2	No risks identified				Thomas Gradel, John T. Nosek	Model of Critical Factors for Outsourcing Agile	AMCIS 2009 Proceedings
						2	Not limited by agile methods				John S. Persson, Bjame R. Schlichter	Managing Risk Areas in Software Development Offshoring: A CMMI Level 5 Case	Journal of Information Technology Theory and Application (JITTA)
	risk agile international	144	100	23.8.2018		2	Not limited by distributed				Sharon Coyle, Kieran Conboy	A case study of risk management in agile systems development	17th European Conference on Information Systems, ECIS 2009
						2	Paper not				Alhua Yan, Vicki Sauter	Risk Management of Information Systems	19th Americas Conference on Information Systems, AMCIS 2013
						2	Not limited by agile				Nipon Parinyavuttichai, Angela Lin	Managing User Requirement Risks – An Exploratory Study of IS Projects from the Views of	KMIS & ConfIRM International Conference 2011 (2011.06)
						2	Not limited				Nipon Parinyavuttichai, Angela Lin	Understanding the emergence of requirement risks in	UK Academy for Information Systems Conference Proceedings 2010
						2	Not identifyi				Linda Bergkvist, Bjöm Johansson	Management of Information Systems Outsourcing:	AMCIS 2011 Proceedings - All Submissions
	risk agile GSD	16	16	23.8.2018	x						Alzoubi, Yehia Ibrahim and Gill, Asif Qumer	Agile global software development communication	PACIS 2014 Proceedings
					x						John S. Persson, Bjame R. Schlichter	Managing Risk Areas in Software Development	Journal of Information Technology Theory and Application (JITTA)

ATTACHEMENT 2.

Data base	Search word/string	Results	Reviewed	Date of search	Double	Criteria 2 fail	Reason 2	Criteria 3 fail	Reason 3	All criteria passed	Writer	Title	Published
AIS Electronic Library (AISeL)	agile requirement risk	1434	100	30.8.2018									
The ACM Digital Library	agile requirement risk	83696	43342							x	Wasim Alsaqaf, Maya Daneva, Roel Wieringa	Agile quality requirements engineering challenges: first results from a case study	2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)
	(+agile +requirement +risk) / matches all	39											
Google Scholar	agile requirement risk	97900	100	30.8.2018						x	B Ramesh, L Cao, R Baskerville	Agile requirements engineering practices and challenges: an empirical study	Information Systems Journal, September 2010, Vol.20(5), pp.449-480
					2		Not about requirements engineering				B Boehm, R Turner	Using risk to balance agile and plan-driven methods	Computer June 2003, Vol.36(6), pp.57-66
										x	I Inayat, SS Salim, S Marczak, M Daneva	A systematic literature review on agile requirements engineering practices and challenges	Computers in Human Behavior 1 October 2015, Vol.51, Part B, pp.915-929
					2		About estimation of implementation risk not identification				P Rempel, P Mäder	Estimating the implementation risk of requirements in agile software development projects with traceability metrics	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 2015, Vol.9013, pp.81-97

Finna database	agile requirement risk	9404	100	43342	2	Not found anymore 27.11.				Tarek El - Najjar ; Imtiaz Ahmad ; Mohammad Al-kandari	Client Communication: A Major Issue in Agile Development International Journal of Software Engineering and its Applications 2016, Vol.10(12), pp.113-130
					2	No only prioritization stuff				Asghar, Anessa Rida ; Tabassum, Atika ; Bhatti, Shahid Nazir ; Jadi, Amr Mohsen	Impact and challenges of requirements elicitation & prioritization in quality to agile process: Scrum as a case scenario 2017 International Conference on Communication Technologies (ComTech)
	(ISD OR software development OR software) AND(risk OR challenge) AND (agile OR continuous)	398	100	43132	2	No 2 not about risks				Anessa Rida Asghar ; Atika Tabassum ; Shahid Nazir Bhatti ; Amr Mohsen Jadi	Impact and challenges of requirements elicitation & prioritization in quality to agile process: Scrum as a case scenario 2017 International Conference on Communication Technologies (ComTech)
IEEE xplore	AND (requirement engineering OR requirements engineering)								x	Rashidah Kasauli ; Grischa Liebel ; Eric Knauss ; Swathi Gopakumar ; Benjamin Kanagwa	Requirements Engineering Challenges in Large-Scale Agile System Development 2017 IEEE 25th International Requirements Engineering Conference (RE)
					2	Not about requirements engineering				Michael Waterman	Agility, Risk, and Uncertainty, Part 2: How Risk Impacts Agile Architecture IEEE Software
					2	About specific RE method				Eduard C. Groen ; Norbert Seyff ; Raian Ali ; Fabiano Dalpiaz ; Joerg Doerr & co	The Crowd in Requirements Engineering: The Landscape and Challenges IEEE Software 2017
									x	Wasim Alsaqaf, Maya Daneva, Roel Wieringa	Agile quality requirements engineering challenges: first results from a case study 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)

