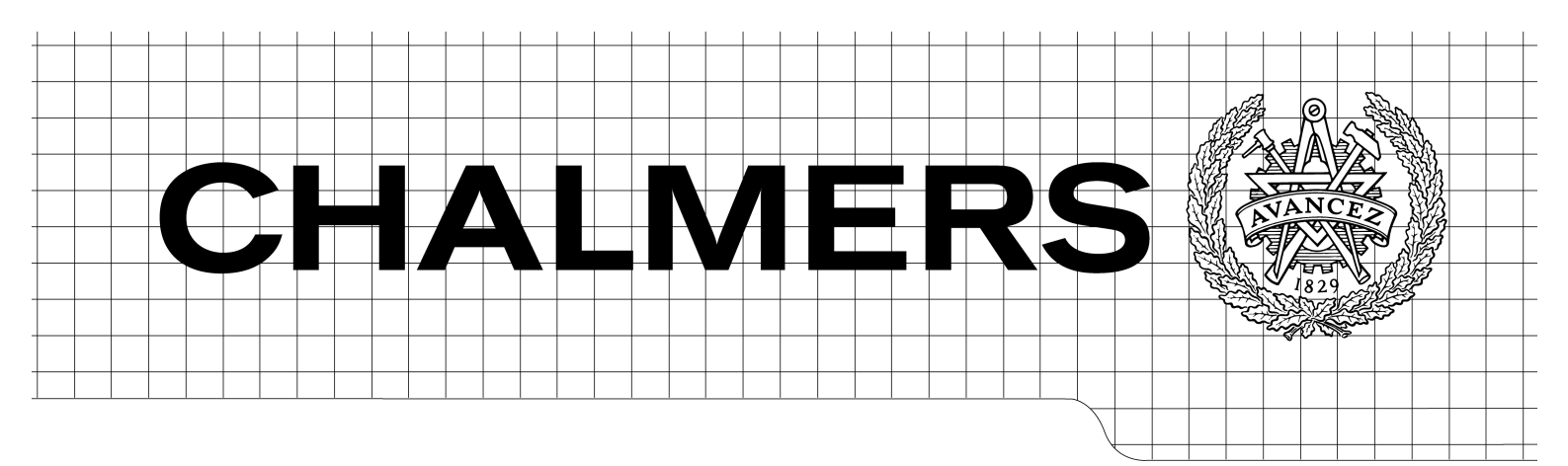

Customer Collaboration in Software Development: A Systematic Literature Review

Master of Science Thesis in Computer Science and Engineering

ANA CONSTANTIN
IRIN MANURASADA

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
Göteborg, Sweden, January 2015

Customer Collaboration in Software Development: A Systematic Literature Review

ANA CONSTANTIN
IRIN MANURASADA

Examiner: RICHARD BERNTSSON SVENSSON
Supervisor: ANA MAGAZINIUS

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
Göteborg, Sweden January 2015

Abstract

Context: Customer collaboration in software development – an industrial perspective.
Objective: To gather an understanding of customer collaboration advantages and disadvantages, problems faced by different organizations when engaging customer collaboration and investigate the processes, tools, guidelines proposed by relevant empirical publications.
Method: Systematic literature review following the Guidelines for Performing Systematic Literature Reviews in Software Engineering
Results: A total of 51 relevant published studies were selected for this study from 6 science and engineering databases. The most discussed challenges of customer collaboration are the lack of stakeholders' commitment, work domain misunderstandings and the lack of methods and processes for collaboration. The most suggested types of solutions are effective and real-time communication tools that help avoid misunderstandings, user group techniques such as meeting and workshops, and user feedback. The most common advantages of customer collaboration are understanding improvement and quick feedback. The most investigated disadvantages of customer collaboration are the reduced in project efficiency and the development of unrealistic expectations towards project roles and the final product.
Conclusion: To be able to gain more advantages from customer collaboration and becoming vulnerable to its disadvantages, a software company should manage the challenges it raises with appropriate solutions.

Table of Contents

1. Introduction

Customer collaboration in software development is seen as a key factor contributing to the project's success [29], [52], [54], [73]. Customers possess domain knowledge of the software being developed, therefore they are an important source of knowledge for developers. Early involvement of the customers can improve design outcomes and promote requirements understanding [53], [67]. Continuous collaboration with the customers throughout the development process has the benefits of adjusting expectations, increasing acceptance rate, and improving system usability [53], [79]. Moreover, continuous customer feedback can help detect any requirements misunderstanding and program defects, and therefore reduce rework cost. In 2001, a group of software development methodologies that incorporated the focus of customers into the entire software development process (SDLC) was promoted. These methodologies are known as agile methodologies [68]. Despite the reported advantages, an approach to achieve an effective customer collaboration is still unclear for practitioners and researchers.

Although agile methodologies encourage larger degrees of customer involvement than other methodologies, they do not always provide explicit means to do so [83], [87]. While methodologies varies in their approaches, customer collaboration in software engineering research also does not receive as much attention as it should. Iivari et al. observed that it is still unclear how to involve customers into system development [73]. Studies published in recent years still reported challenges in customer collaboration such as the lack of customer commitment and the gap between the customers' and developers' knowledge [18], [29]. With more and more companies embarking on global software development, global software projects seem to suffer more due to differences in culture, and time zones [9], [18], [59].

In order to gain a clearer understanding on how industrial practitioners can effectively collaborate with customers, the first step is to examine current state of empirical evidence on this topic. The objective of this thesis is to investigate empirical studies on customer collaboration in order gather evidence of challenges, solutions, and advantages along with disadvantages of customer collaboration in software development. Therefore, a systematic literature review (SLR) was chosen as the methodology [76].

The study was conducted in three stages: planning, conducting, and reporting. The guidelines stipulated by Kitchenham and Charters [76] was followed in order to ensure rigorous process and unbiased measures. The primary studies, i.e. relevant research publications, which addressed empirical evidence in an industrial setting was investigated more closely than other settings. It is so that the review results could be applicable for industrial practitioners. In this thesis, two groups of project stakeholders are concerned: customers and development team practitioners. A customer is a role of a stakeholder in software development project who provides financial funding, and/or, produce software requirements, and/or use the final software product. A development team practitioner role refers to software project leaders, software developers, or project members with technical background. From the search in 6 science and engineering digital libraries, a total of 51 primary studies were synthesized and analyzed to quality assess and answer research questions identified.

The contributions of this thesis are twofold. For academic area, this research can be used as a starting point of a new research in the topic of customer collaboration by taking a closer look

into the most common challenges faced by customer collaboration. For industrial use, the results of this research provides a list of recommendations for practitioners. The recommendations are aimed for the practitioners to choose the right tools and methods in order to improve the quality of collaboration. Also, to reduce drawbacks it may inflict the management of the project.

This thesis is structured as follows. Section 2 presents Background and Related Work. It summarizes findings in customer collaboration before looking in depth at the topic and motivates the purpose of this research. Section 3 describes methodology. It presents the setup of this research and discusses validity threads to the study. Results are presented in section 4. It contains an overview of the primary studies and answers the research questions. A discussion of results are presented for each one of the research questions. An overall discussion is presented in section 5. The Conclusion in section 6 summarizes the findings of this paper, the areas for further research, as well as to give an overall perspective on the topic of customer collaboration.

2. Background and Related Work

2.1 Background

In this thesis, the definition of customer collaboration refers to the one given by Cockburn and Highsmith as "... actively working together to deliver a work product or make a decision" [64]. Customer collaboration has been known to have a positive influence on software system success [29], [52], [54], [73]. Therefore, the topic of customer collaboration can be found in software development studies since the past decades.

In older studies, such as the ones during 1970 – 1990, customer collaboration in software development were often focused mainly during the initial stage of the SDLC as suggested by Iivari et al. [73]. Kraut and Streeter [78] mentioned that approaches suggested in these studies tend to base on the assumptions that requirements do not or cannot be changed once the project has moved to the implementation stage. The communication format used tended to be formal, i.e. via a formal meeting sessions. In 1970, Royce [86] suggested customer review sessions during the following stages: requirements gathering stage, critical software review before coding stage, and final software acceptance review stage. The highest amount of customer collaboration effort was on the critical software review stage. The review should be in-depth and continuing. Later in 1982, Gladden [69] argued that iterations in the waterfall model was not an efficient way to develop software. His proposal to solve the problem still focused on customer involvement prior to implementation stage. Gladden proposed to have concrete agreements of project objectives with the management of the customers since the beginning of a project. At the same time, early presentation and testing of software mockups could improve the customers' understanding of what the end result should be. With these two propositions, Gladden claimed they could reduce the problem of customer changing requirements during implementation stage, hence reduce the rework on the code [69].

With the introduction of Agile Manifesto [68] in the year 2001, the value of customer collaboration has become more critical to software development projects [81]. Agile is a group of software development methodologies that have different approach in customer collaboration than Gladden's. Fowler and Highsmith [68] suggested that instead of stating customer requirements contractually at the beginning of the project, the development team could better understand the true needs of the customer with ongoing collaboration. Change to requirements are welcomed, even at the later stage of a project. This way the project team will be better placed to deliver a product that really matches customer's needs. This process is evident in Extreme Programming (XP) where customers are required to participate throughout the project. Since the beginning of an iteration, customers choose the highest priority user story to be implemented first. Hence, customers can control the planning and the developers do not have to worry about correct prioritization. During iterations, customers sit together with the development team full-time and clarify any ambiguous requirements or questions that developers may have [56]. Customers are also encouraged to write test cases, ensuring thus the required system operability [56]. These means of collaboration can be seen to help reduce misunderstanding between customers and developers and enhance customer satisfaction in the product. As opposed to the older methodologies, Coram and Boner stated that agile projects tend to favor informal communication as it is easier to transfer tacit knowledge [66].

However, agile methodologies alone are not enough for practitioners to efficiently collaborate with customers. While most agile methodologies defined roles and responsibilities for customers in a software project, only XP explicitly specified how to collaborate. In other methodologies, it is understood that frequent feedback from customers is vital, but the frequency are not clearly presented. Also, details of physical proximity between customers and developers is only evident in XP. A probable explanation could be due methods such as Scrum or Lean was designed to be a lightweight framework [83], [87]. They only indicate what should be done, but allows the freedom for practitioners to incorporate other techniques and processes to achieve it [64].

Despite the methodologists' effort to involve customers into software development, academic studies still report barriers that prevent successful collaboration as well as conflicting evidence on the effectiveness it has on a project. Bruce et al. [18] reported that customer collaboration can lengthen the development process and be difficult to control. Financial concerns such as customer collaboration adding cost to project budget, failing to return investment, and causing project overruns were reported [18], [39]. Commitment barriers were revealed in studies by Cockburn, Boehm, and Coram and Bohner [57], [63], [66]. That is, it is difficult to establish relationship with customers in a way that they are willing to collaborate throughout the project. Especially with XP, where on-site customer is postulated. Boehm [57] also observed that customers might not be committed to collaboration or insufficiently informed to be able to collaborate effectively when the software company attempts to use an agile development process [57]. The recent trend of global software development [61] imposes challenges due to cultural, time zone, and communication constraints [9], [18], [59]. Also, Tait and Vessey's [89] study found that positive relationship between the extent of user involvement and system success was not statistically significant.

2.2 Related work

Customer collaboration has been seen in the related literature of this study to have both a positive and a negative effect on the project success. In their mapping study, Abelein and Paech [52] find an overall positive relationship between customer collaboration and software project success. This overall positive influence is also found by Bano and Zowghi in their systematic literature review on customer collaboration relationship with software project success [54] and on their systematic literature review concerning customer involvement in requirements engineering and its connection to project success [55].

However, despite the majority of findings in these literature review studies which indicate an overall positive correlation between project success and customer collaboration, a low but significant percentage indicates in all three related studies that customer collaboration might also negatively influence project success. Abelein and Paech [52] find that 10% of the studied publications indicate a negative effect of customer collaboration on project success. In Bano and Zowghi's systemtic literature review [54], 8% of the studied publications have shown negative influences of customer collaboration on project success and 24% have yielded uncertain results. Although Bano and Zowghi [55] found no negative influence of customer collaboration on requirements engineering, 4 out of 13 studied publications yielded uncertain results. Moreover, regarding the design phase of a software product development, only 9 studies found positive results, whilst 3 showed negative results and 6 reached uncertain results. This is also the case of the implementation phase of software product development, where 1 study showed negative results and another one uncertain results, compared to 5 studies reaching positive results [55].

Besides looking into the correlation between project success and customer collaboration, Abelein and Paech [52] have also looked during their systematic mapping study into means of increasing customer collaboration proposed in the studied literature. The proposed practices for improving customer collaboration have been classified in their research into the following categories, which correspond to project activities and phases: management, specification and requirements engineering, design and implementation, validation, as well as evolution. Nevertheless, these aspects have not been analyzed in depth due to the systematic map method used. Bano and Zowghi [55] have also researched the factors to be taken into consideration for an effective customer collaboration during requirements engineering, but this was done mostly from the perspective of user's psychology while other influencing factors have not been taken into consideration.

Therefore, it is conflicting weather adopting customer collaboration during a software development process leads in all cases to positive results [52], [54], [55]. Even though attempts with the purpose of investigating means of achieving a successful customer collaboration have been done, more empirical evidence in this direction is needed. This thesis extends these researches by exploring the subject of customer collaboration during the entire development process and conducting a broader analysis on the subject in details. Therefore, the aim of this systematic literature review is to analyze the advantages and disadvantages of customer collaboration during software product development while attempting to examine them from the perspective of the challenges an effective customer collaboration implies and means of overcoming challenges in order to achieve product success.

3. Methodology

The systematic literature review method allows for the identification, assessment and combination of evidence found in primary empirical studies regarding customer collaboration. As the evidence existing in the primary studies might be contradictory, their analysis offers a broad perspective on the researched subject. Therefore, it has been chosen for this study as its purpose is to summarize existing industrial evidence on customer collaboration advantages, disadvantages, challenges and means of overcoming this challenges in software engineering projects and produce a conclusion regarding customer.

The process of conducting the systematic literature review in this paper was done following the "Guidelines for performing Systematic Literature Reviews in Software Engineering Version 2.3" by Kichenham and Chartes [76]. The review comprises three main phases detailed in Figure 3.1 - planning, conducting and reporting.

3.1 Planning the review

This sections details the activities conducted during the first phase, planning the review, of this systematic literature review. During this phase, the following activities were performed: defining research questions, developing review protocols, and validating the review protocols.

3.1.1 Research questions

The research questions of this study are as follows:

RQ1 What are the challenges of customer collaboration found in the studied literature?

RQ2 What are the solutions/methods/processes/tools proposed in customer collaboration found in the studied literature?

RQ3 What are the advantages and disadvantages of customer collaboration found in the studied literature?

3.1.2 Develop a review protocol

The review protocol development step includes: developing data extraction form, identifying databases to search on, formulating a search string, and identifying the criteria for inclusion and exclusion.

Data extraction form

After the research questions were formulated, the data extraction form was designed. The form consists of four parts: general information (such as reviewer names and review date), information for the general overview of the publications (such as the type of publication, methodology used in the study), information for the quality assessments of the papers, and information for answering the research questions. The form was designed to be broad as it cannot be anticipated a priori what kind of data might be encountered. The properties which are to be extracted from the form are listed in Table 3.1. Each property is linked to an objective which can either be a research question or general information about the studies which is considered helpful in this research. More details of the form can be found in Appendix A.

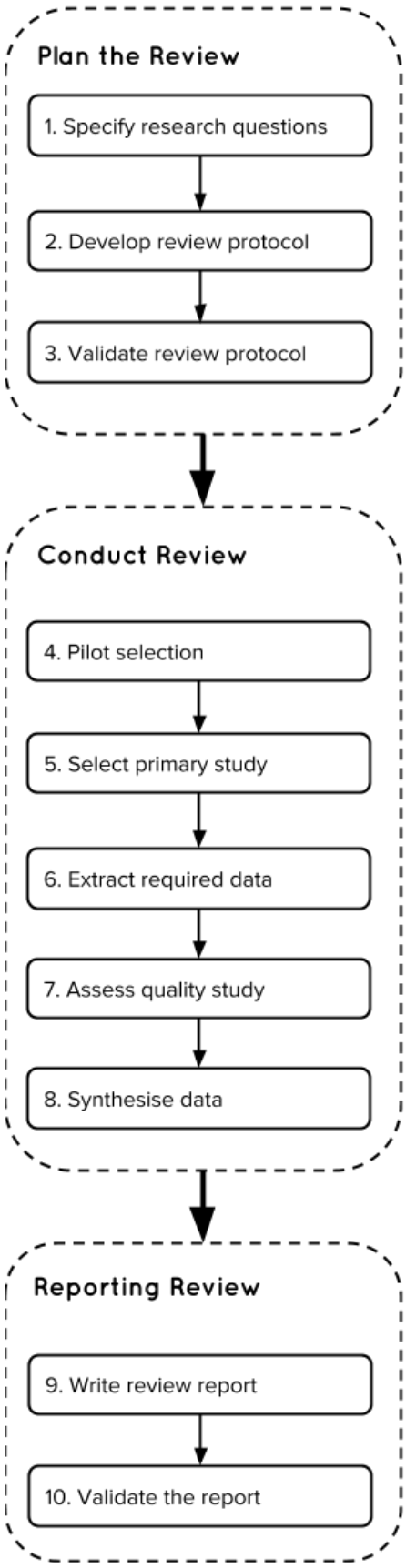

Figure 3.1 The systematic literature review process for software engineering used in this study was modified from [76], [84]

Table 3.1 Extracted properties

ID	Property	Objective(s)
P1	Publication year	Overview of the studies
P2	Publication type	Overview of the studies
P3	Methodology	Overview of the studies
P4	Data collection method	Overview of the studies
P5	Industry	Overview of the studies
P6	Size of the company	Overview of the studies
P7	Timeframe of the study	Overview of the studies
P8	Participants	Overview of the studies
P9	Software development process	Overview of the studies
P10	Location	Overview of the studies
P11	Activities	Overview of the studies
P12	Proposed tools, framework, process	RQ2
P13	Challenges	RQ1
P14	Advantages	RQ3
P15	Disadvantages	RQ3

Digital libraries

Six digital libraries were selected based on the recommendations in the guidelines [76], namely:

1. IeeeXplore
2. ACM Digital Library
3. Scopus
4. Engineering Village: Inspec and Compendex
5. ScienceDirect
6. SpringerLink

Search strings

From the research questions the search term was first identified as "customer collaboration software." Then, the synonym of the search term was found to be "customer involvement software." This resulted in below search string:

"customer AND (collaboration OR involvement) AND software"

The search string was used to query publications by titles and abstracts except with SpringerLink, where abstract search is not available. Here, the search string was applied to titles and context instead.

To evaluate the quality of the search string, pilot searches were conducted on Scopus and SpringerLink. The abstracts of the search results on the first pages were checked for the relevancy to the subject. As most of the abstracts were found relevant, it was persuasive that the search string was credible.

Even though the related SLR had used a broader search string in their reviews by including synonyms such as "enduser", "involv*", "participat*", "contrib*" [54], [55], the preliminary search in this study yielded more results, 7211 papers, on the common used databases (IeeeXplore, ACM Digital Library, ScienceDirect and SpringerLink), compared to 965 papers

found by Bano and Zowghi [54], [55]. These differences can be seen in Table 3.2. One possible reason for this discrepancy is that the preliminary research of this study was done both in the title and in the abstract of the papers, while Bano and Zowghi's [54], [55] search string was limited to the abstract.

SpringerLink has constituted an exception in this study as the website does not allow for customized abstract search, thus the search has been performed in the title as well as the entire paper and it has yielded a number of 6,312 papers found on the initial search and 14 primary studies to be analyzed. Considering the relevant difference in the number of papers found during Bano and Zowghi's [54], [55] SLRs – 43 papers on initial search and 2 primary studies to be analyzed – we can deduce their search did not extend to the entire paper when only title search was allowed by the database.

The terms "participat*", "contrib*" have not been seen as relevant terms to be included in the search string because they do not refer specifically to customer collaboration, thus they would not yield concluding results. Furthermore, at a search on Google Scholar, "customer involvement" yielded 823,000 results and "customer involv*" yielded 4 results. Therefore, the term "involvement" have has been used as an alternative for "involv*". The term "enduser" has not been used as an alternative for "customer" because the purpose of this paper is to look into the way software companies collaborate with their customers, which include the entire customer organization, not only software end-users.

Considering the relevant difference in the number of papers found during the database search in this study compared to the number of papers found in [54] and [55], the search string in this paper has been kept to the specified minimum. Moreover, it will be seen further on in this chapter, 31 studies have been selected for final analysis in this paper, compared to 9 in [54] and [55] for the 4 common used databases. These differences can be seen in Table 3.2.

Table 3.2 takes into consideration the results of the search strings of this paper and the ones in the SLRs of Bano and Zowghi [54], [55]. Bano and Zowghi used the same search strings and the same results for both their systematic literature reviews.

Table 3.2. Results comparison for the common used databases

	Preliminary results	Selected papers for final analysis
Current study	7,211	31
Studies [54] and [55]	965	9

Concerning the additional databases used in the related literature (Citeseer x, MISQ and Google Scholar) [54], [55], researchers have agreed to use instead Scopus and Engineering Village (Inspec and Compendex). The reason behind this is the fact that the later databases are engineering oriented and not management oriented such as Citeseer x and MISQ. As the purpose of this paper is not the study of customer collaboration influence on project success, but to investigate the challenges and solutions in customer collaboration. Publications in the engineering field are of interest to this research because they allow for the analysis of specific software engineering processes concerning customer collaboration.

Google Scholar was not used in the final review protocol due to practicality reasons. When trial searched with the defined search string in titles, Google Scholar returned only 13 results. Among the 13, only 5 publications were relevant and they were already covered by other

digital libraries. However, when searched in abstracts, the database returned over 400,000 results, which was too large to manage during the time frame.

The results of the search string in this paper have not been compared to the results of Abelein and Paech's [52] mapping study because of the difference in the research method. Moreover, as they state in their study, the search string used is similar to the one used by Bano and Zowghi [54], [55].

Inclusion and exclusion criteria

For filtering the research papers found during the search through the databases, inclusion/exclusion criteria were formulated.

For the inclusion criteria, the reviewers agreed to include:

- All papers with empirical evidence on customer collaboration or involvement in industrial setting.

As for the exclusion criteria, the reviewers exclude the results contains at least one of the following characteristics:

- The papers that are non-software development related study.
- The papers that are non-journal or non-conference articles.
- The papers that are not written in English.
- The paper's subject is not related to customer collaboration.
- The paper is a theoretical study.

The inclusion/exclusion criteria are to be applied first for titles of the studies found during the database research. Following, the included papers will have their duplicates eliminated and the inclusion/exclusion criteria will be applied to their abstracts. The process will be repeated for full text reading. The remaining papers will be collected for in depth analysis and completion of the data extraction form. This process is explained in section 3.2.

3.1.3 Validating review protocol

The two researchers have agreed upon the protocol developed for conducting this systematic literature review. As mentioned in section 3.1.2, each item was piloted by the two researchers and modified accordingly. For validation purposes, the research setup was validated by the thesis supervisor for validation prior to the conducting the literature review process.

3.2 Conducting the review

Figure 3.2 shows the number of publications found during the database search and how many remained during each step of the elimination/inclusion process, as well as after it.

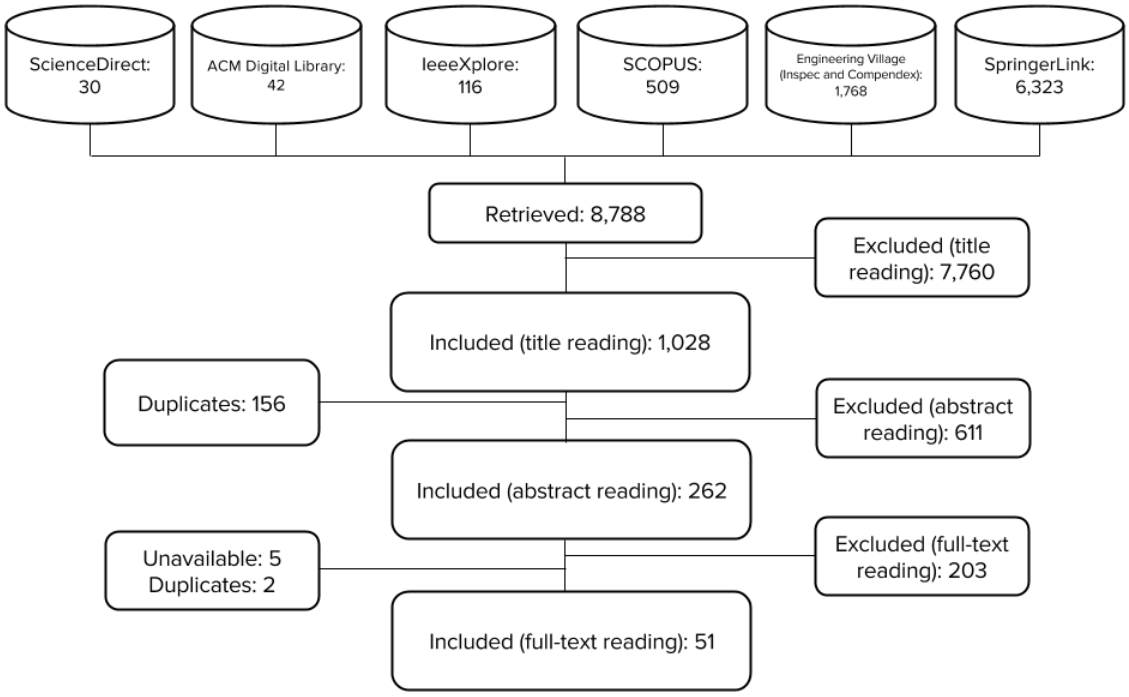

Figure 3.2 Primary studies selection

Five papers were not available through the electronic and library resources, thus they have been excluded from the full-text reading. Duplications were detected and removed using the functionalities in EndNote and Microsoft Excel.

3.3 Reliability of inclusion decisions

Before proceeding with publications' selection, pilot tests were conducted to measure and adjust the level of agreements between the two reviewers.

The pilot tests were performed to ensure that the inclusion and exclusion criteria can be reliably interpreted between the two reviewers [76]. Based on the guidelines' recommendation [76], a Cohen's Kappa statistic was used to measure the level of agreement [8]. For this research, the documented Strength of Agreement by Landis and Koch [80] was used as the benchmark of an acceptable reviewers' level of agreement.

The process researchers followed during the title exclusion/inclusion pilot test consists of the following steps:

1. Randomly select 50 publication titles
2. Each reviewer applies inclusion/exclusion criteria to the titles independent of each other
3. Compare the selection results
4. Calculate Kappa statistic
5. Compare the statistic to "the strength of agreement"

6. If the strength falls on the "almost perfect" range (Kappa statistic result between 0.81 – 1.00 [80]), then discuss and resolve the mismatched results and proceed with the publication selection. Else, investigate the causes for disagreement, solve them, and repeat the process from step 1.

For the first pilot, 50 publication titles were randomly selected from the search results yielded by IeeeXplore database. The appropriate sample size for the pilot was not mentioned in Kichenham and Charters' guideline but the Cochrane handbook suggests 10 to 12 samples should suffice [70]. However, considering the large number of papers retrieved for this review, extending the sample size to 50 has been viewed as a more reliable measurement. The results of the assessment are presented in Table 3.3. From this, the Kappa statistic yielded 0.75 which according to Landis and Koch [80], falls on the "substantial" range of the strength of agreement. The cause of the disagreement was the misinterpretation of vague terms such as "Jazz and Eclipse collaboration." One reviewer who was not familiar with these technologies included the titles for further refinement at the abstract selection stage. The reviewer who was more familiar with the terms explained and the disagreement was resolved.

Table 3.3 Reviewers' agreement on the first pilot test

		Reviewer 2		Totals
		Included	**Excluded**	
Reviewer 1	**Included**	38	2	40
	Excluded	2	8	10
	Totals	40	10	50

Since the first measurement result did not fall under the "almost perfect" range, the process from step 1 was repeated, until the mismatch was resolved, as follows.

Other 50 randomly selected titles from IeeeXplore were assessed. The results are presented in Table 3.4. The Kappa statistic obtained was 0.919, which falls under the "almost perfect" range according to Landis and Koch [80]. The disagreements were discussed and resolved.

Table 3.4 Reviewers' agreement on the second pilot test

		Reviewer 2		Totals
		Included	**Excluded**	
Reviewer 1	**Included**	22	1	23
	Excluded	1	26	27
	Totals	23	27	50

With the desirable level of 90+% agreement achieved as a result of the Kappa statistic, the reviewers were able to divide the work and apply the inclusion/exclusion criteria to the remaining titles.

Forty-five papers, corresponding to 5% of the included papers after title elimination process, have been selected for the abstract elimination process. Both researchers decided to include the papers they were not sure about. The Kappa test yielded a 0.952 agreement level. The researchers disagreed on just one paper which was later discussed and excluded.

Prior to the whole paper elimination process, researchers performed a new Kappa test for 15 of the papers. The researchers were in complete agreement regarding the exclusion/inclusion of papers based on the whole paper reading.

The data extraction form filling was done for all the papers in parallel by both researchers. Results were compared afterwards and differences were discussed until a consensus was reached on the information to be extracted and noted from each research paper.

3.4 Study quality assessment

Regardless of their qualitative or quantitative study type, all primary studies used in this review have been quality assessed. The purpose of this assessment is to ensure that the claims of the reviewed papers fall under an acceptable credibility level and to provide recommendations for further research [76].

The assessment data was extracted in the data extraction form as mentioned in section 3.1.2. There are six questions in the data extraction form as described in Table 3.5. QA1, QA2, QA3, QA4, and QA5 were ranked among three choices: Yes, Partially, and No, where Yes means the item is clearly stated, Partially means the author stated but some questions remained and No means the item are not stated at all. QA6 was also ranked by three choices Yes; Yes, and the paper position towards it; and No.

Table 3.5 Quality assessment results

ID	Quality Assessment Question	Yes	Partially	No
QA1	Objectives or research goals are clearly stated	94.12%	1.96%	3.92%
QA2	Research questions and/or hypotheses are clearly stated	50.98%	5.88%	43.14%
QA3	Threats to validity or limitations identified	47.06%	1.96%	50.98%
QA4	Was the sample size justified? (quantitative studies only)	78.57%	7.14%	14.29%
QA5	How clear and coherent is the reporting?	90.20%	9.80%	0%
ID	**Quality Assessment Question**	**Yes, and position towards it**	**Yes**	**No**
QA6	Related work/background/literature review described	78.43%	9.80%	11.76%

The results show that 96% of the publications have research objectives or goals stated clearly (QA1). However, only approximately 50% clearly stated the research questions and/or hypotheses, while 6% partially stated them and 43% did not state at all (QA2). One of the reasons leading to a relatively high percentage of papers not stating the research questions/hypothesis could drive from the fact that the papers are industrial reports.

One major concern on the quality assessments of the reviewed publications are the validity threats (QA3). It is shown that 47% of the reviewed literature contains validity threats or limitations section, while approximately 5% do not.

Table 3.6 illustrates the ratio of validity threat types found in the reviewed literatures which mentioned validity threats or limitations. The majority of the validity threats specified are concerned with external validity, especially the threat to generalizability with 42%. This is because it is inherited from the case study research method, which is the most used method

among the reviewed publications. The Other type includes project specific limitations such as the proposed tool is in an explorative stage, the author lacks domain experience, etc. The "personal bias" threat to validity have been specified 9% of the research papers. This bias indicates that either the author has participated in the experiment or the interviewees were prone to trying to guess the answers so that they suit the study objective due to personal interests.

Table 3.6 Types of validity threads

Type of validity threat	Frequency	Percentage
External validity	14	42.42%
Other	10	30.30%
Bias	3	9.09%
Construct validity	3	9.09%
Small sample size	3	9.09%

Sample size justification (QA4) is developed for assessing the quality of quantitative primary studies, as sample size is vital to the credibility of the results especially for those studies that make statistical inference. It is shown that 78% of the quantitative primary studies justified their data size.

To assess the coherence (QA5) of the reviewed publications, the reviewers ranked the coherence on a scale of Yes, Partially, and No. "Yes" reflects the papers that are coherent. "Partially" is used to rank papers that ideas are coherent but not so correlated or structured. "No" is used to rank papers that are not coherent and ideas are not connected at all. The results reveal that 90% of the reviewed literatures are clear and coherent, while 10% are partially clear and coherent. No paper has received a No ranking. This could be the result of the exclusion criteria that does not allow the inclusion of papers which are not journal or conference articles. Therefore, all included primary studies have an acceptable level of coherence.

Related work and background (QA6) is another criterion for quality assessing the investigated studies. It shows if the authors of the reviewed publications have performed prior background research on the subject and reporting the background properly. The papers properly reporting the related work and background correspond to "Yes, and position towards it" ranking. The lack of related work and background in around 12% of the studies might be attributed to industrial studies.

Overall, the quality assessment reveals that the studied literature is of an acceptable standard, however, researchers should pay more attention to the validity threats in order to make the research in the field of IT and software engineering corresponding to a high scientific standard.

3.5 Validity threats

An SLR should include all the relevant studies to answer a research question in a repeatable way. In order to achieve this the researchers had followed the guidelines by Kitchenham and Charters. However, some deviation from the guidelines were made in the interest of research practicality and the time management of the review research and management report. The following threats to research validity are identified:

1. Review protocol,
2. Reporting bias,
3. Bias in researcher's role as reviewer, and
4. Quality reporting bias.

3.5.1. REVIEW PROTOCOL

A threat to validity of this SLR is caused by the relevant studies identification in the review protocol. The following elements of the review protocol are identified.

A. Exclusion criteria,
B. Identification of synonyms, and
C. Identification of publication source.

A. Exclusion Criteria

The exclusion criteria excludes all the publications that are non-English, non-journal, or non-conference proceedings.

Language: This restriction may cause relevant studies in non-English languages to be excluded. This restriction was necessary, however, because English is the only language both researchers have in common.

Journal and Conference: The exclusion of non-journal and non-conference publications is intended to exclude studies that are not fully researched or peer reviewed. A possible effect of this exclusion is that studies published as workshop that may be relevant were not searched.

Workshop Restriction: Workshops tend to be exploratory in nature. Results of tools or methods evaluated in workshops are likely to be preliminary in content and probably inconclusive. Reports from workshops were therefore excluded from the review.

B. Identification of Synonyms

The use of too few synonyms in the search string could result in relevant studies that use other synonyms being excluded from the study. However, adding more synonyms returned too large results. An example is the use of 'user' as a synonym for 'customer', SpringerLink gave 26,858 results for 'user' which is too many results to filter. Jorgensen and Shepperd also illustrated a similar issue in [74]. Considering research time constraints, we chose to use a search string with fewer synonyms but which produced adequate results.

C. Identification of data source

The review protocol restricted the searches to the predefined science and engineering digital libraries. This may cause relevant studies published on other fields to be excluded from the review. An example is shown by Bano and Zowghi who for their research found 56 relevant studies in business and management fields [54], [55]. To reduce data source limitations this SLR searched all publication venues available in each digital library. The use of Scopus and Engineering Village (which [54] and [55] did not use) provide a wide range of publication sources in fields as diverse as medical, manufacturing, and business management. More details of the rationale was elaborated in section 3.1.2.

3.5.2. REPORTING BIAS

Reporting bias may occur if the dissemination of research findings is influenced by the nature and direction of results [70]. The bias can be reduced by extending the search into 'grey literature', unpublished studies, and by searching on less popular databases [76]. Such extensions may cover more results with negative reports. This SLR used an automatic search strategy reputable science and engineering databases only. Therefore the reporting biases that may affect this SLR are: publication bias, duplicated publication bias, language bias, and location bias [70]. Language bias with English is unavoidable as previously mentioned. Duplication publication bias has been mitigated as both researchers extracted data of all primary studies individually. Any duplication would be detected during the full-text data extraction.

A. Location Bias

Location bias occurs when studies with positive findings get published in 'easy access journals' or in popular databases. The researchers did not extend the research searches to less well known databases. They reduced location bias by including conference proceedings in research searches.

B. Publication Bias

Publication bias are bias that occur when studies with positive findings get higher publication exposure than studies with negative findings. In this SLR publication bias was mitigated by using the same inclusion criterion to select all primary studies and by not specifically searching for positive finding studies. Still, publication bias could affect the results of RQ3, as there may be more reported advantages of customer collaboration rather than disadvantages. The impact of this bias is low since the aim of RQ3 is not to prove that customer collaboration is beneficial but to summarize and list advantages and disadvantages reported in the relevant studies. Publication bias could also affect RQ2 if tools and methodologies are sponsored by influential groups. For this SLR the background context of sponsorship was identified if available.

3.5.3. BIAS IN RESEARCHER'S ROLE AS REVIEWER

Decisions and interpretation may be biased by a researcher's previous knowledge which leads to information anchoring, misunderstanding, and inconsistency in the interpretation of the review protocol. Although individual judgment cannot be totally without subjectivity, this threat is considered to be low as it followed the researchers followed Kitchenham and Charters's guidelines' and have no 'conflict of interest' in the research finding.

The guidelines suggest that an SLR should be conducted by more than one reviewer and follow a predefined protocol [76]. Two researchers conducted this SLR with a supervision from an experienced researcher in software engineering. Although on some parts of the process the researchers worked separately to filter publications, the inter-rater agreement was measured until the "almost perfect" range was achieved [80]. This SLR was conducted following a predefined protocol. It was developed by both of the researchers, piloted, and later on verified by the experienced researcher in software engineering. This methodology ensured that both researchers have the same understanding on how to use the protocol and how to use the data extraction form.

No Conflict of Interest

The researchers do not have personal interests in producing either positive or negative findings. None of the primary studies is authored by any of the researcher conducting this

SLR and there is no personal interests influencing the researchers' opinions. Also, the researchers do not receive any compensation based on the direction of the findings.

3.5.4. QUALITY ASSURANCE REPORTING AMBIGUITY

To minimize the effect of the bias in researcher's role as reviewer on quality assessment, a check list was designed to be objective. Each primary study was checked to see whether the quality existed, partially existed, or did not exist. It is noted that although something is not reported in a primary study it may in fact have been done [76]. If necessary this ambiguity could be clarified by contacting the primary study authors but it is not necessary since the majority of the primary studies selected have already received good ranking as can be seen from Table 3.5. Adding more 'Yes' to the table, would not significantly change the overall results. Quality Assurance reporting ambiguity risk in this SLR report is considered to be low and adequate for the purpose of this research.

4. Results and analysis

This section reports and analyses the findings on customer collaboration based on the extracted data from the primary studies. Subsection 4.1 gives the overview of the primary studies used to conduct the systematic literature review. As each primary study was conducted in a different background setting, some data needed for the research data extraction form such as company size or software development methodology used is not 100% available in all the literature. As a consequence of these data deficiencies the analysis and discussion that follows is of necessity generalized. Subsections 4.2, 4.3, and 4.4 are the results and discussions of RQ1, RQ2, and RQ3 respectively.

4.1 Overview of the study

Section 4.1 provides an overview of the reviewed publications in terms of publication trend, publication types, and methodology used. The overview provides the background setting for the research questions findings and analysis.

Publication trend

Figure 4.1 shows that the publications that were reviewed were published between 1989 and 2014. The number of relevant publications increased from 2000 until 2007. After that the trend drops until it reached its peak of 7 publications in 2011. This corresponds to the introduction of agile methods which promote customer collaboration as one of its core values. From 2011 the new publications trend fluctuates.

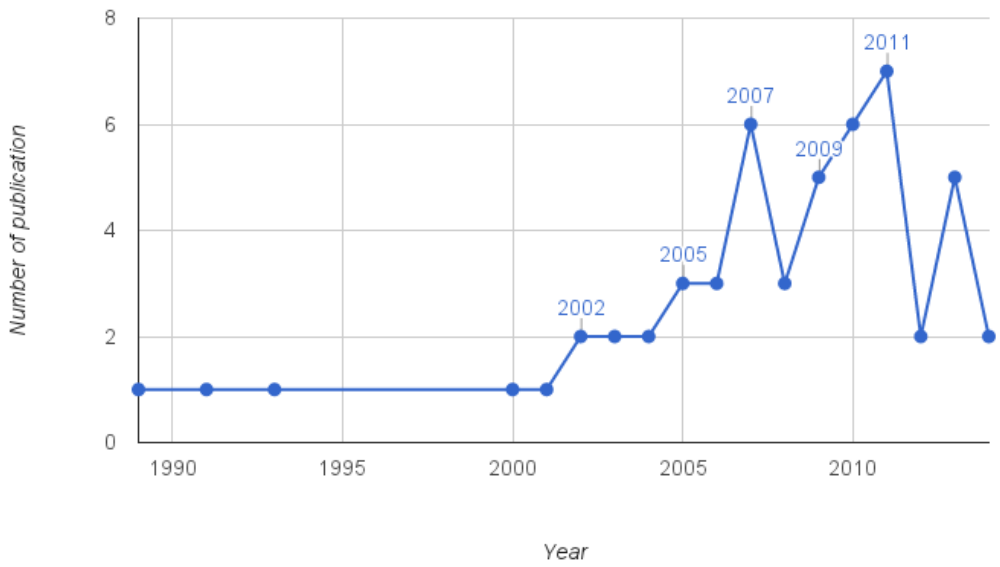

Figure 4.1. Publication trend

Publication type

The primary studies reveal that 58.8% of the reviewed literature are journal articles, while 41.2% are conference articles, as can be seen in Figure 4.2.

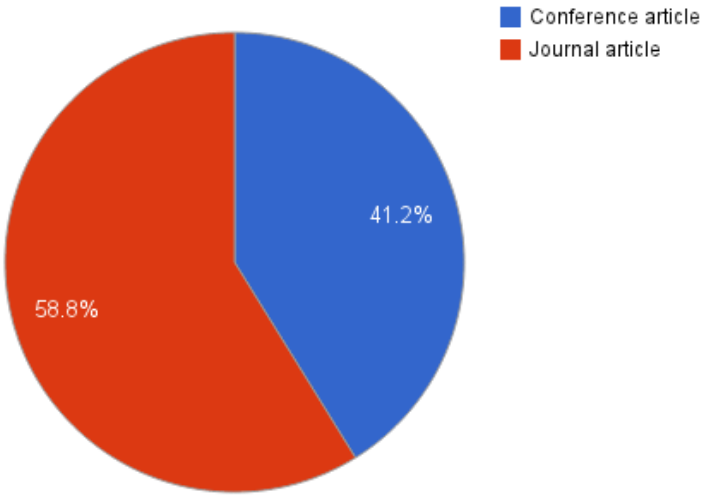

Figure 4.2. Publication type ratio

Methodology

The methodology used the most among the primary studies is case study, 45%. The second most used is a tie between survey research and industrial report of approximately 16% each. Industrial report is a type of research in which the author(s) do not use any scientific methodologies but rather take an active part of the development process and reports the experiences from an industrial perspective. The different distribution in methodologies among the researched studies is presented in Figure 4.3.

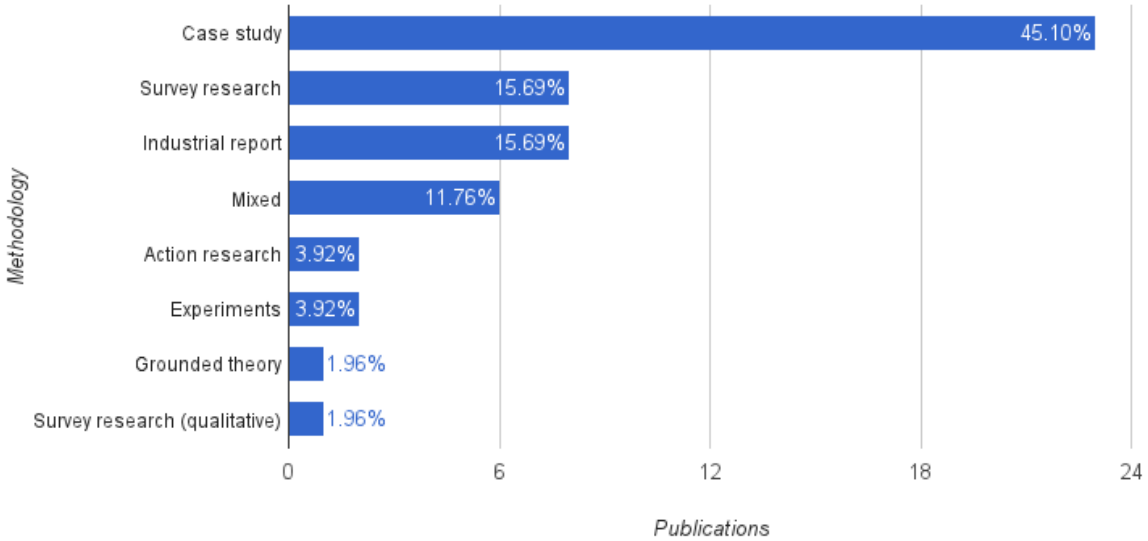

Figure 4.3. Distribution of methodology

Data collection method

The majority of data collection methods used in the primary studies are questionnaires which contributes about 18% of the total publications. Interviews and publications that do not stated the data collection method used is about 14% each. Most of the publications that do not state data collection method belong to industrial reports. The remaining are a mixtures of different types for instance interviews complement with observation sand documentation review.

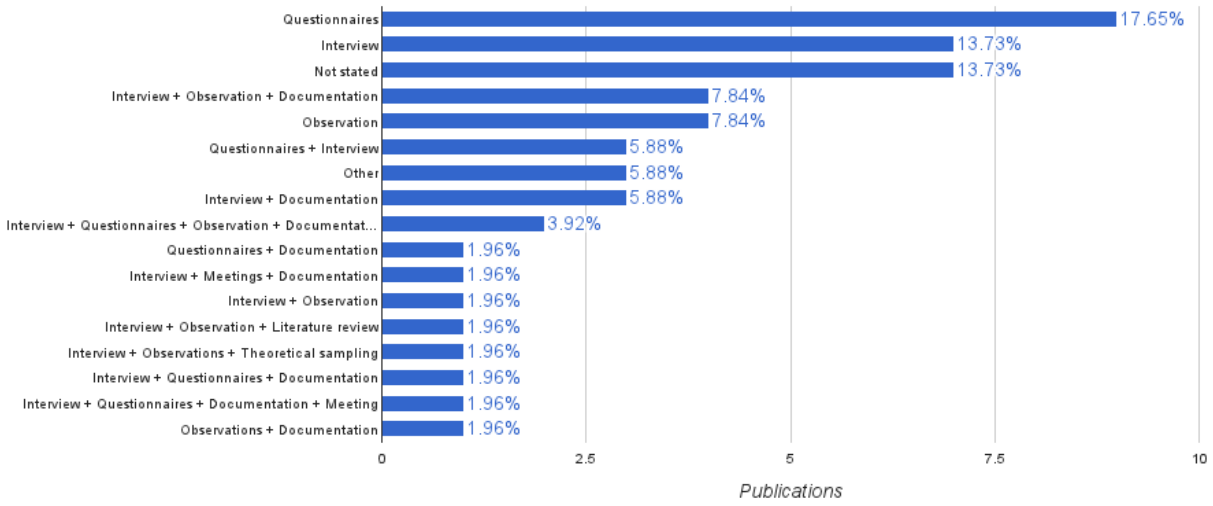

Figure 4.4 Distribution of data collection method

Software development method

Most of the publications do not state the software development method in the context of the study, while those that do tend to be agile methodology. That is, 16% mentioned agile but do not specify what type of agile the company being studied is using. XP has the distribution of 8%. Plan-driven method is used in only 4% of the selected studies and one study uses both agile and plan-driven methods.

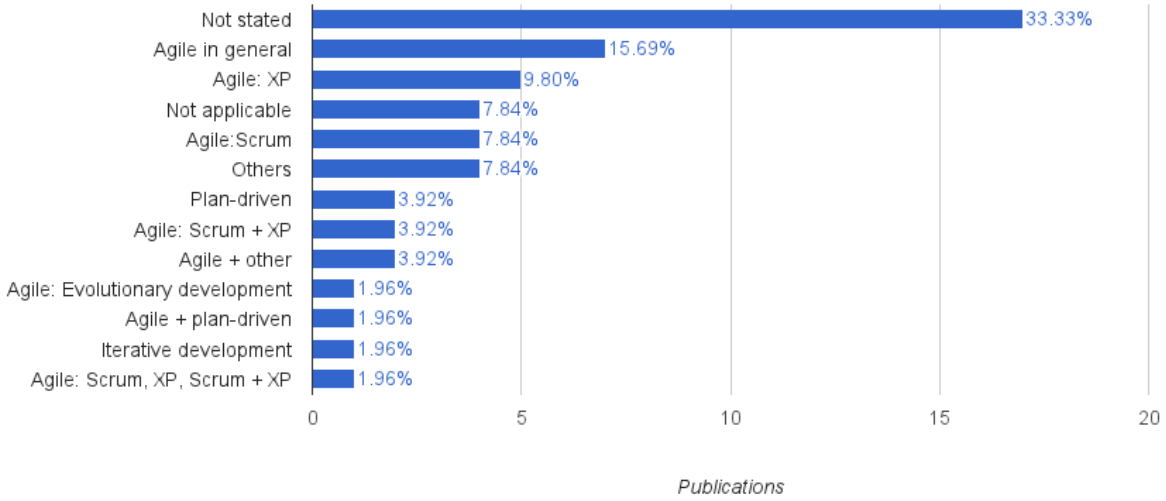

Figure 4.5 Distribution of software development methodology used in the primary studies

Physical proximity

Almost 55% of the primary studies do not state the context of the proximity between the software development team and the customers. Among the papers that do, 32% report long distance or distributed collaboration. Approximately 6% of the primary studies researched both on-site and long distance collaboration, while on-site collaboration itselft is almost 4%

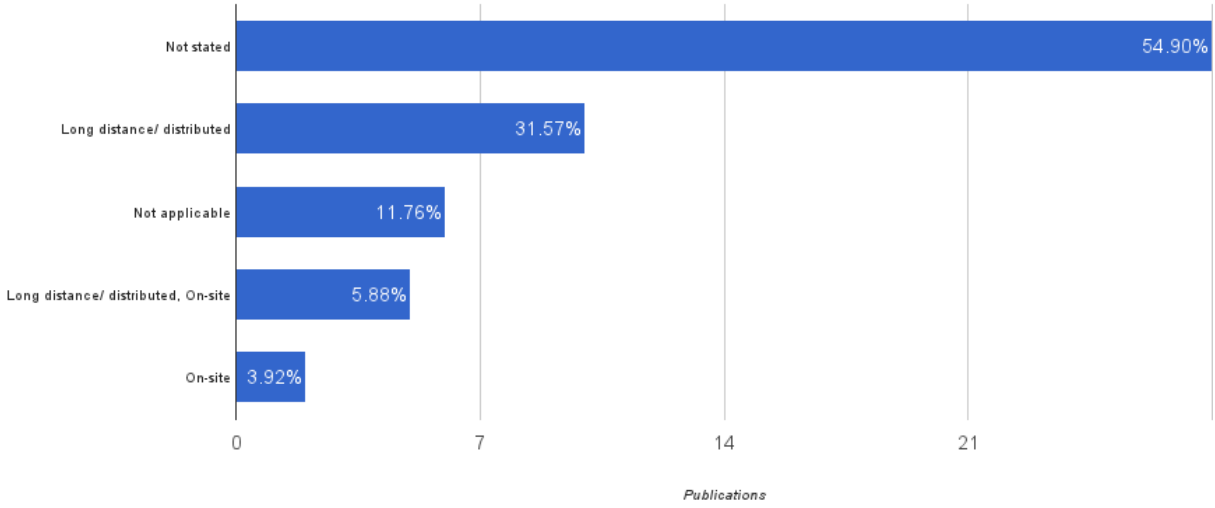

Figure 4.6 Distribution of types of physical proximity

Activities

Approximately 39% of the primary studies discuss collaboration during the entire development process. Almost 18% studied collaboration in requirements engineering activities. About 8% do not state any activities. Those reports tend to emphasize the importance of collaboration. Same distribution of 8% is for design phase, while about 6% mentioned collaboration during the implementation phase of the software development.

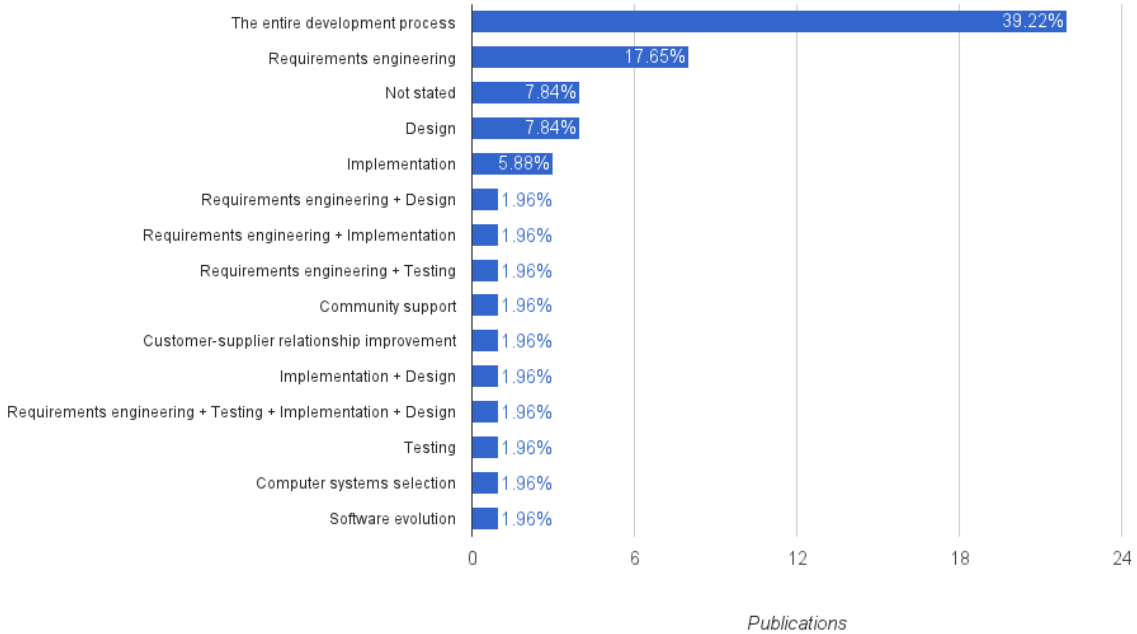

Figure 4.7 Distribution of activities

Participants

Primary studies that stated the role participants in the studied project tend to involve more than one role in a study. Developers represent the majority of the participants followed by customers and users.

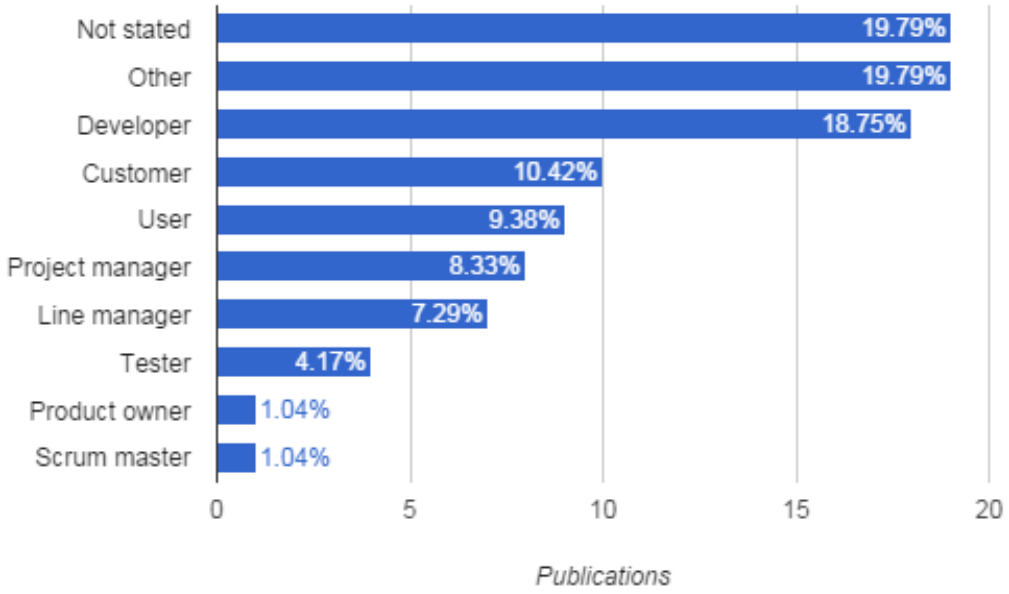

Figure 4.8 Distribution of participant roles

Industry

Almost 69% of the primary studies focused on a software development company. There are studies that specifically study the collaboration between a customer of a specific industry customers and software development teams. Customers' industries include social welfare, space science, biomedical science which are industries that are considered to require complex software and a deeper understanding of domain knowledge prior to software development.

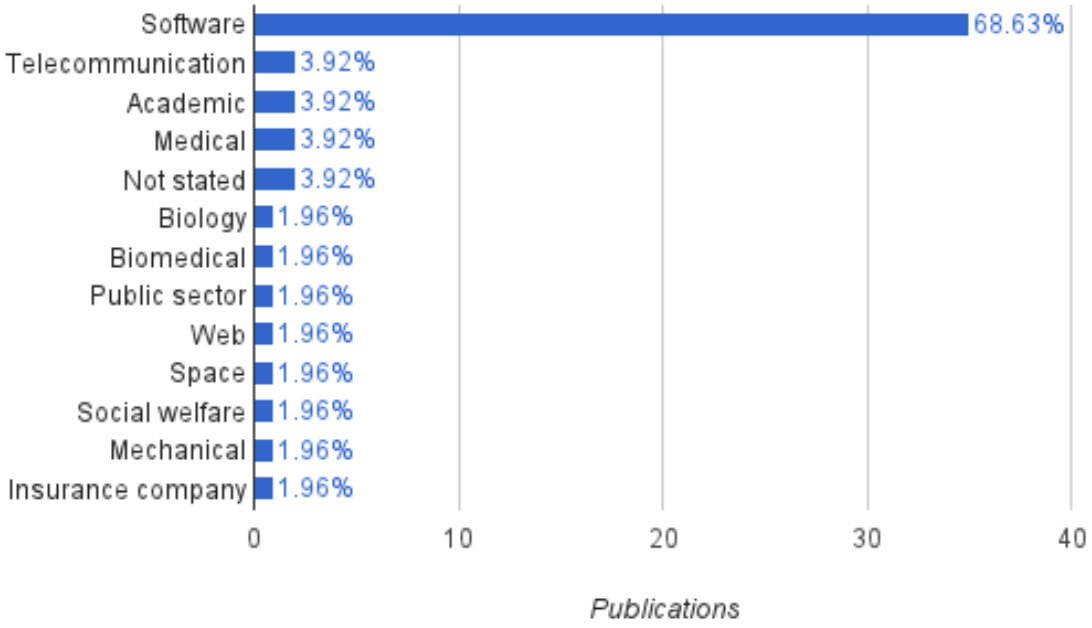

Figure 4.9 Distribution of industry

Company size

About 39% of the primary studies do not mention the size of the company being studied. For the studies that do state the size of the company, slightly more than 25% are large companies, almost 10% are medium size. Small and a mixture of company sizes in one study account for about 4% each.

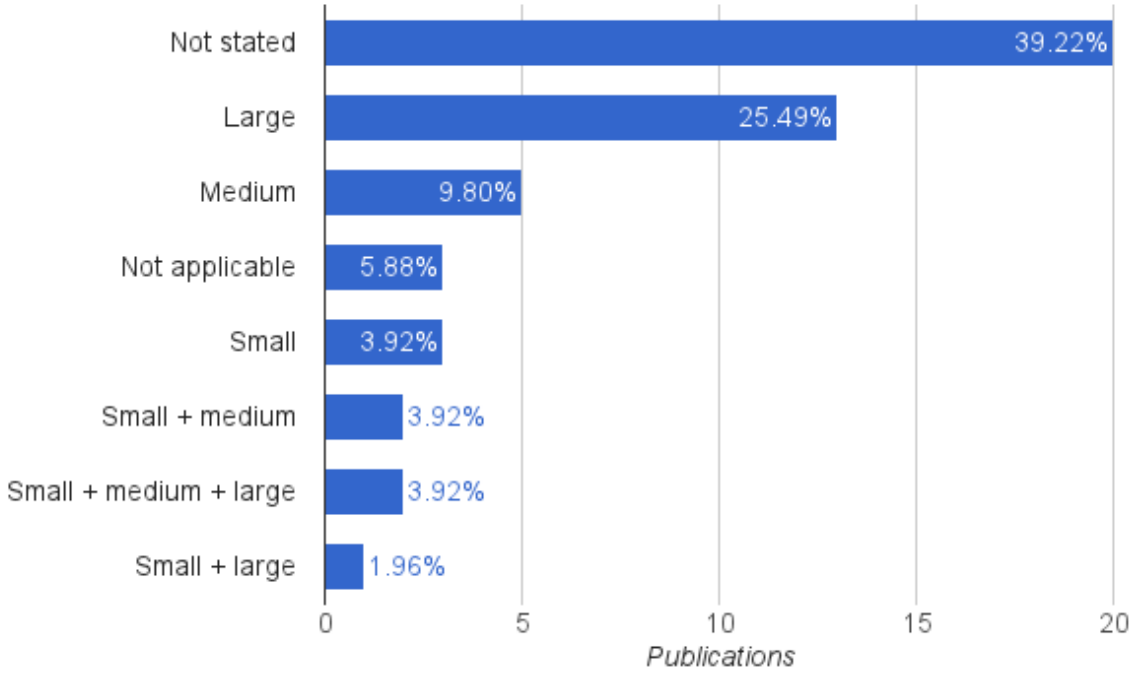

Figure 4.10 Distribution of company size

Time frame of the study

Most of the primary literature, especially industrial studies, does not state the time frame of the study. Those that do state the time frame are conducted during a period of less than 1 year. The longest time frame of a study was 4 years.

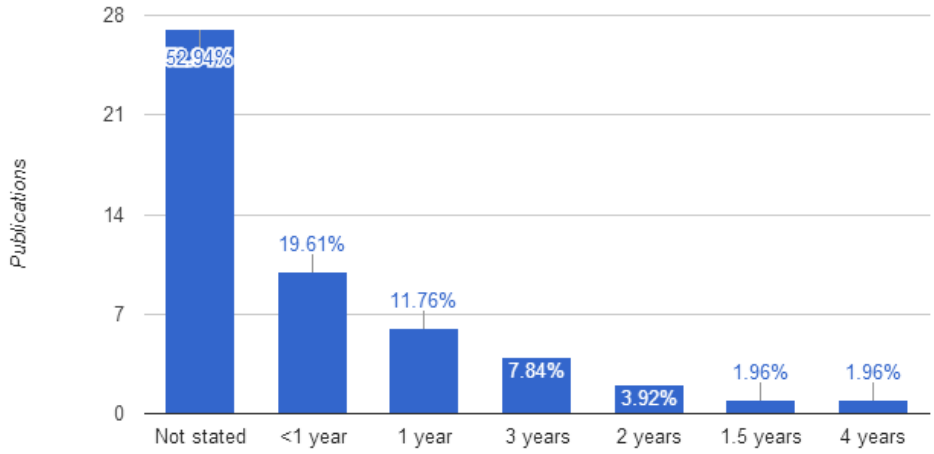

Figure 4.11 Distribution of time frame of study

4.2 What are the challenges of customer collaboration found in the studied literature? (RQ1)

Results

Approximately 48% of the 51 primary studies discuss challenges that developed during customer collaboration. Almost 75% of the publication discusses the customers and their environment as a cause of communication challenges during collaboration. Roughly 54% of the research publications discuss the challenges faced by the development team when trying to engage in a customer collaboration process, and 42% of the papers discuss challenges for both customers and developers during collaboration. The challenges for both customers and developers include commitment, trust, criticism and decision-making processes. The breakdown of these components is shown in more detail as Table 4.1 below.

Table 4.1. Challenges frequency based on topic

Number	Topic	Papers	Frequency	Percentage
1	Challenges related to users and their environment	[6], [9], [11], [13], [16], [18], [19], [20], [25], [29], [32], [36], [37], [38], [39], [40], [41], [42]	18	75%
2	Challenges for both developers and users during the collaboration process	[5], [9], [18], [24], [29], [31], [32], [36], [39], [48]	10	42%
3	Challenges faced by the development team alone	[5], [9], [12], [18], [24], [26], [29], [31], [32], [37], [38], [39], [40]	13	54%

The major challenges found in the primary studies are found to be the lack of commitment on both customers and developers. This can cause from the lack of the awareness to collaborate, or the lack of personal skills or support to commit to the collaboration despite wanting to do so. The second mostly discussed challenges are the lack of understanding of each other's work fields. Customers are not familiar with development processes and developers do not have a clear understanding of the customers' field of work. The factors found to prevent collaboration during this SLR are listed in Table 4.2.

As shown in Table 4.2, the tabulated extraction of the challenges discovered in the primary studies revealed similar challenges in multiple studies and more frequently in a software company attempting an effective customer collaboration process. Other challenges are analyzed by only a few of the primary studies but they should not be disregarded and should become the subject of future research.

Table 4.2. The most important challenges frequency

Topic	Challenges	Papers	Frequency	Percentage
1	Stakeholders lack commitment and personal skills, as well as environment support	[5], [9], [18], [24], [27], [29], [31], [36], [39], [48]	10	38%
1	Customers not familiar with common techniques (agile, prototypes) used in software engineering	[9], [11], [18], [32], [36], [38], [40]	7	27%
3	Developers are not familiar with customers domain problems	[11], [25], [36], [39], [42]	5	19%
2	Geographical dispersion	[9], [12], [18], [31], [38]	5	19%
2	Lack of time for commitment	[18], [29], [38], [40]	4	15%
3	Lack of methods and practices for customer communication/collaboration	[5], [29], [32], [40]	4	15%
1	Stakeholders do not have a clearly defined terminology	[6], [37], [42]	3	12%
2	Communication weaknesses on delivering and presenting the information	[32], [36], [39]	3	12%
2	Technical means of communication	[9], [27], [36]	3	12%
3	Dealing with large number of customers	[18], [24], [40]	3	12%
2	Cultural differences	[9], [31]	2	8%
1	Users don't participate in the requirements prioritization sessions	[16], [20]	2	8%
1	Misconceived view on roles of stakeholders	[13], [19]	2	8%
2	Reduced level of trust	[5], [9]	2	8%
1	Change resistance from users	[39]	1	4%
1	Having customers perform too many tasks such as backlog management may create cognitive overhead.	[41]	1	4%
1	Stakeholders reluctance in premature sharing of their data (competition is fierce)	[42]	1	4%
1	Few involved users might be active users	[16]	1	4%
3	A lot of customer criticism might arise when involving the customers in early development phases	[24]	1	4%
2	Long-distance: ineffective decision-making meetings and difficulty in managing conflict, as well as having a common understanding of requirements	[9]	1	4%
3	Requirements must take into account the needs of different users and stakeholders with different interests	[38]	1	4%
3	Planning for customer specific teams is difficult	[37]	1	4%
3	The costs might outweigh the potential for sales	[39]	1	4%
3	Fixed-bid contracts	[18]	1	4%
3	Onsite customer can make developers feel disturbed.	[26]	1	4%

Discussion

The following discusses the most frequently discussed challenges in customer collaborations: stakeholders' lack of commitment to collaborate, the gap in technical or domain knowledge, and geographical dispersion.

1. Stakeholder's lack of commitment to collaborate

The lack of commitment from software organization's staff and customers is found to be the most commonly reported challenge in customer collaboration. It is found that 50% of the primary studies discussed this challenge were conducted in a long distance setting [9], [18], [27], [31], [48], while the remaining of the studies did not mentioned the level of collaboration proximity. The studies discussed that, long distance collaborations cause difficulties that may hinder stakeholders from collaboration commitment. Causes of such difficulties can be categorized into: difficulties in establishing trust, difficulties in time allocation, and the lack of support.

A. Difficulties in establishing trust. It is difficult to establish trust between customers and developers in a long distance setting, thus they can be reluctant to collaborate. Primary study [9] emphasizes on the importance of trust building via informal communication as one of important factors in successful collaboration. The authors found that members in a long distance project lack the opportunity to personally know each other via informal communication, such as coffee break talk, which leads to the lack of trust and finally resulted in the lack of commitment.

B. Difficulties in time allocation. When it is difficult to conduct long distance collaboration, customers can be less likely to allocate their time to commit to the development project. It can be implied from [9] and [18] that due to the long distance nature, the customers were not required to be actively involved. Since it is not required, customers would give higher priority to their daily routine tasks rather than development project tasks.

C. Lack of support. The lack of support can be categorized in to, the lack of tool support and the lack of customers' organization support.

C.1 Tool support: Study [31] reported that the inability to commit to long distance collaboration could come from the lack of support to use collaboration tools. Development companies in [31] did not receive adequate after sale support from the collaboration tools manufacturer. Hence, developers were unable to commit to the collaboration during the downtime of the tools.

C.2 Customers' organization support: Primary study [48] is a research from Oracle that reported its difficulty in maintaining customers' commitment throughout the project. This collaboration is done long distance with multiple customers companies, which Oracle could not demand for their total commitment.

2. The gap in technical or domain knowledge

The second mostly discussed challenge is customers' lack of technical knowledge. This is closely related to the third mostly discussed challenge which is developers' lack of domain knowledge. Approximately 57% of the primary studies reporting customer's insufficient technical knowledge were conducted in agile context [18], [32], [38], [40]. It is said that agile was chosen for these type of projects because it provides an environment for close working relationship between customers and developers [25]. On the other hand, the challenge of developers' lack of domain knowledge were found in the industry that required expert domain knowledge such as biomedical science [25], social welfare [39], government sector [36], and biology [42]. These challenges can be categorized as follows:

A. Customers do not understand agile process
B. Customers do not understand the domain knowledge they are representing
C. Customers are unable to communicate domain knowledge
D. Both customers and developers lack the understanding of one another

A. Customers do not understand agile process. Customers who came from industries other than IT generally do not have a priori knowledge of agile methodology [32], [36], [38]. The unfamiliarity of customers towards agile is shown in primary study [18]. The authors investigated the cases where customer representatives were inefficient in giving developers requirements on time. The lack of agile understanding also resulted in the customers' inability to prioritize requirements. This leads to the inefficiency for the developers as they have to wait for the input or need to rework due to incorrect requirements [18].
B. Customers do not understand the domain knowledge they are representing. In some agile methodologies, customer representatives are the intermediary between the end-user and the developers. In the cases where customer representatives are not the end-user themselves, they might not have the full understanding of the end-user's need [18], [32], [36], [40]. Thus, resulting in ambiguous requirements that required rework from developers.
C. Customers are unable to communicate domain knowledge to developers. Some customers came from industries that required specialized domain knowledge [25], [42]. It might be difficult for them to translate their tacit knowledge into common terms. Also, in some industries, customers may have difficulties revealing confidential data that might cause them to lose competitive advantage. Such resulted in ambiguous requirements for developers [25], [42].
D. Both customers and developers lack the understanding of one another. Without sufficient knowledge of one another's field, effective communication can be difficult to obtain. Some same terminologies or ideas might have different meanings in different industries, causing misinterpretation in requirements [25], [32], [42]. For instance, study [25] mentioned the words "database" in software engineering and "data base" in biomedical science refer to different ideas despite pronouncing the same way. To resolve this challenge, developers might consider attending trainings or conducting a research on business domain knowledge. However, financial aspects must be taken into consideration for such activities. If the training or research cost is higher the potential sales of the software product, financial damage can occur to the project [39].
Customers from various industries requires developers' technical knowledge to create quality software. Agile methodologies seem to be an appropriate method for collaboration, however it is natural that knowledge gaps exists. The challenges discussed have shown that it is vital to lessen the knowledge gap between customers and developers in order to create quality requirements and efficiently develop the software.

3. Geographical Dispersion

Geographical dispersion is also the third mostly discussed challenge in customer collaboration along with developer's lack of domain knowledge. It has been discussed in 20% of the primary studies that reported challenges in collaboration. The most common context of the studies reported geographical dispersion as a challenge is the use of agile methodology (60%) [12], [18], [38]. The Agile Manifesto [68] stated that "face-to-face communications is the best form of communication." Also, agile activities such as acceptance testing and frequent communication are easier to be done in close proximity. Thus, agile projects conducted in a long distance collaboration can be more challenging than those conducted in co-located space. The causes of geographical dispersion challenges in agile context found in the primary studies are the need for collaboration tools and the complexity in securing customers feedback.
A. Need for collaboration tools. The difficulty in establishing technical infrastructure for software sharing is a cause of geographical dispersion challenge [12]. In agile methodology, customers are required to periodically test the working software and give feedback to developers. Pichler et al. [38] reported that testing is more difficult to conduct in long distance setting. In order to achieve customer feedback in a long distance setting, various

communications and collaboration tools are needed. Setting up telecommunication infrastructure such as teleconference systems, screen sharing software, or software sharing via application service provider (ASP), is required in a long distance setting [12]. Acquiring such infrastructure imposes higher financial investment. However, this is unavoidable as customer's feedback is reported to be valuable for developers to create quality software and increase their confidence [12].

B. Complexity in securing customer feedback. Customer feedback is more complicated in a long distance setting [18], [33], [38]. Study [9] showed that customers who are closer to the development team tend to collaborate more as the meetings or communication can be easily established and less formal. If the customers are close by, developers can easily create an ad hoc request for feedback, however this is more difficult in global collaboration with time zone difference [18]. Similarly, [33] reported that long distance collaboration cannot guarantee customers' continuous availability for questions. Moreover, [38] agree that communication during long distance collaboration needs to be as uncomplicated as possible in order to assure quality requirements.

4.3 What are the solutions/methods/processes/tools proposed for solving the problems in customer collaboration found in the studied literature? (RQ2)

Results

Among the total of 51 primary studies, 35 of them (68.6%) discuss solutions, methods, processes, and tools for solving problems in customer collaboration.

These 36 studies have been further classified into four areas: tools, complete process guidelines, general suggestions and client-consultant relationship model[1]. Among 14% of these studies proposed tools to be used during customer collaboration. Over 29% of the studies discuss specific suggestions for solving the problems during customer collaboration. The rest of the publications, 57%, present general suggestions useful in enhancing customer collaboration. The results can be seen in the Table 4.3.

Table 4.3. Types of suggestions proposed and their frequency

Number	Suggestion for solving problems in customer collaboration	Papers	Frequency	Percentage
1	Tools	[1], [11], [15], [30], [50]	5	14%
2	Specific suggestions	[5], [19], [21], [24],[34], [37], [41], [45], [48], [51]	10	29%
3	General suggestions	[3], [4], [6], [7], [8], [9], [10], [14], [16], [18], [20], [24], [27], [31], [32], [33], [36], [38], [43], [46]	20	57%
4	Client-consultant relationship model	[13]	1	3%

The proposed tools, as well as the specific suggestions are summarized in the following sub-sections and detailed in Appendix B' Table B.a and Table B.b, respectively.

[1] One research paper presents customer collaboration suggestions from the perspective of hiring consultants, so it has been included as a separate area as the situation is found to be different from engaging regular customer collaboration [13].

General suggestions have been divided into two categories, suggestions based on the methods to be used and suggestions based on the process. Table 4.4 and Table 4.5 list the suggestions found in the primary studies regarding the methods and the processes to be used for an effective customer collaboration.

Among the general methods suggested, communication tools which allow for immediate and effective response is mostly suggested, followed by changing customer mindset. Most tools suggested are to promote synchronous communications, e.g. instantaneous, real-time voice or video tools rather than asynchronous. For the general process suggestions, the use of user group techniques is the most commonly suggested solution. It is the technique refers to communication techniques which allow for immediate and effective response such as interviews, workshops, and meetings. The second most suggested process is customer feedback and iterative development.

Table 4.4. Frequency of most common method related general suggestions found in the primary studies

Topic	Suggestion	Papers	Frequency	Percentage
3	Provide communication facilities/tools to customers for problem solving to be on time and without misunderstandings	[9], [16], [18], [31], [36]	5	25%
3	Change customer mindset towards participation	[14], [18], [20], [38], [46]	5	25%
3	Provide knowledge to customer so that they can describe the problems better.	[36], [38]	2	10%
3	Site visits, schedule face-to-face kick-off meetings at the beginning of global projects	[9], [32]	2	10%
3	Invest time to make sure the development team understands and agrees on who the target audience is	[33]	1	5%
3	No need for the customers to work in the same room as the developers, but they should be on-site	[27]	1	5%
3	Offer adequate, friendly and professional support	[16]	1	5%
3	The requirements engineer should be the only link between customers and development team	[38]	1	5%
3	Enhance management support	[14]	1	5%

Table 4.5. Frequency of most common process related general suggestions found in the primary studies

Topic	Suggestion	Papers	Frequency	Percentage
3	Use user group techniques such as workshops, interviews, requirements gathering sessions, etc.	[4], [6], [7], [32], [33], [43]	6	30%
3	Use customer input in a development environment through feedback	[4], [7], [10], [16], [38]	5	25%
3	Use iterative development	[3], [4], [7], [8], [38]	5	25%
3	Demonstrating prototypes/demos to users prior to development	[7], [8], [9], [18]	4	20%
3	Get input from specific customer studies	[4], [8], [24], [33]	4	20%
3	Making sure ongoing requirements review and approvals are received from users	[7], [36]	2	10%

3	Customer involvement in early stage of documentation	[3], [10]	2	10%

Discussion

The following discusses the implications of the most frequently reported general suggestions, specific suggestions, and tools suggestions.

1. General suggestions

General suggestions are suggestions that do not give any specific named methods, processes, or tools to solve issues in customer collaborations. They are categorized into general suggestions of methods and general suggestions of processes.

A. General suggestions of methods

A.1 Provide effective communication tools. Practitioners are suggested to provide the development team with suitable communication tools. These tools are aimed to facilitate a better, on-time communication, which does not leave room for misunderstandings in problem-solving pursuits [9], [16], [18], [31], [36]. However, these papers do not provide suggestions for specific tools. Primary studies that provided these suggestions are found to be in a long distance context [9], [16], [18], [31]. Primary study [9] provides suggestions to have communication tools to enhance informal communications between different locations. These tools can encourage trust between customers and developers. A database that collects customers feedback is an important tool in distributed collaboration environment as suggested in [16]. Primary study [18] suggested that E-collaboration tools are very important in ensuring regular customer feedback. Primary study [31] provides a list of communication tools features desired by developers to help aid their long distance collaboration. The top three features are voice, desktop sharing, and ease of use, which are considered to be synchronous communication characteristics.

A.2 Change customers mindset. As one of the most common challenges consists in stakeholders' commitment, changing the customer mindset towards participation is impetuous. Sixty percent of the primary studies that provided this suggestions were conducted in agile context [18], [20], [38]. Customer feedback is very important for agile, therefore it is not surprising to see more proportions of papers suggesting that customers commitment should be secured. This could be achieved by changing the perception of project importance and benefits, as well as ease of participation [46]. Moreover, enhancing management's support [14] could be a push towards increasing developers' commitment and customers' trust which in turn makes customers more willing to participate. Providing customers with knowledge [36], [38] could increase comprehension of the development process, thus their ease of participation.

B. General suggestion of process

B.1 Employ user groups technique. The most frequently mentioned process related suggestions are the suggestions to apply a user group technique in customer collaboration. They are the processes to ensure quality feedback or requirements from the customers. The majority of primary studies (67%) mentioning these process suggestion are conducted in agile context [4], [32], [33], [43]. This technique consists of activities such as workshops, interviews, and requirements gathering sessions. In general, the tenchinques are to be incorporated into the agile methodology that the development teams are currently using. Primary study [4] suggested sessions for requirements gathering and quality assessments to not only ensure working software, but also to help detect and avoid problems in ethical and

social values. Primary study [32] stated that user group is a means of cooperation that developers preferred for gathering user needs and requirements. This is due to better understanding of the system from the end-users. Primary study [33] suggested processes such as interview or focus groups sessions to gather customers' opinion, feelings, and goals of the project. Primary study [43], also mentioned that during such activities, governance responsibilities should also be discussed in order to avoid problems of miscommunication, lack of ownership, and lack of coordination in the iteration.

2. Specific suggestions

Most of the specific suggestions (30%) are meant to be used to complement an agile methodology to enhance the methodology's effectiveness. The proposed specific suggestions can be further categorized into specific process and specific role suggestions.

A. Specific process suggestion

One primary study proposed a specific process suggestion. The proposed suggestion is called Strip Resolution Process (SRP) [45]. The SRP is based on the ethnographic process and is intended to lessen the understanding gap between the customers and developers in requirements engineering. The authors proposed steps that practitioners can follow in order to validate that the developers' understanding is correct. This process is to be used at the beginning of each iteration in agile methodology.

B. Specific role suggestion

There are two primary studies that proposed specific role suggestions. A customer-specific team [37] is a design to respond to the customer's needs and requests after product deployment. This role can improve long-term relationship with the customers and reduce cycle time. It also provides short feedback loop, which aligns with the agile concept.

The last role proposed is the role of the story master [41]. A story master is a role to help reduce workload of an agile customer. The objective is to allow the customer to have more time to perform more important tasks such as requirements prioritization and not as much on smaller tasks such as backlog management or writing acceptance tests. The authors claim that by transferring less important tasks to a story master, the project could have economic benefits. However, the selection of the personnel who would assume this role has to be done carefully.

3. Tools suggestions

The majority of the suggested tools are the tools to be used in design and requirements engineering activities [1], [11], [15], [30]. Within the proposed tools 75% are web-based applications, while 25% proposed modelling notations. Moreover, 75% of the tools proposed were designed or conceptualized by the authors of the primary studies. The studies were conducted to observe the usability and effectiveness of the tools in industrial setting [11], [15], [30]. The remaining of the proposed tool is the existed commercial tool [1]. The study was conducted to investigate the author's research questions on end-user involvement when applying the tool. The proposed tools can be further categorized by their aims as: Tools for post-development phase collaboration, and Tools for development phase collaboration.

A. Tools for post-development phase collaboration.

One primary study conducted a study on a social media tool that encourages customers to provide requirements to improve an existed software product [1]. The authors conducted the study using a commercial tool called Get Satisfaction. It is a web-based platform where software companies can create a supportive community for its customers and developers. Via

this tool, the customers can post their ideas on the existing product and later on developers can reply to it. The customers' post can be translated to a requirement which developers can implement into a shared product that all customers can use. This primary study illustrates a possible usage of social media to receive feedback and requirements from customers for a continuous improvement after the software was launched.

B. Tools for development phase collaboration.
The remaining of the primary studies suggest tools that are to be used during the development phase of a software product [11], [15], [30]. They are all designed or conceptualized by the authors of the primary studies. The common characteristics of these tools are the ease of use for the customers and to reduce misunderstanding in communication. For user interface design activities, DigiPap is a web-based tool that the customers can drag and drop design elements to create a software prototype themselves [11]. The prototype can be shared with developers and feedback could be made throughout the development process. For user interface and software modelling activities, User Interface Transition Diagram (UITD) was proposed by [15]. The authors claimed that these modelling notations are different from existed modelling notations such as UML in a way that it can improve the communication between the customers and the developers. UITD supposes to "give information for all parties but without the party-specific information that could generate confusion." For software requirements management activities, primary study [30] proposed a web-based tool called RM-Tool. Similar to [15], RM-Tool also has modelling notations that is not too technical for the customers while gives enough technical information for developers. Other than modelling notations, the tool also has features such as secured connection, requirements traceability, and priorities and responsibility allocation. The authors claimed that, such a requirements management collaboration tool can improve project management, communication, and understanding.

It can be seen that most of the tools proposed by the primary studies are web-based tools, hence can be utilized in a long-distance software development context.

4.4 What are the advantages and disadvantages of customer collaboration found in the literature? (RQ3)

Results

Among the 51 reviewed publications, 59% investigated the advantages of customer collaboration, while disadvantages have been discussed in 25% of the papers. The advantages and disadvantages of customer collaboration have been analyzed based on how they influence different project aspects. The project aspects are categorized into 12 aspects such as project success, quality of work, efficiency, etc. The overview of the comparisons between advantages and disadvantages on different aspects is illustrated in Figure 4.11.

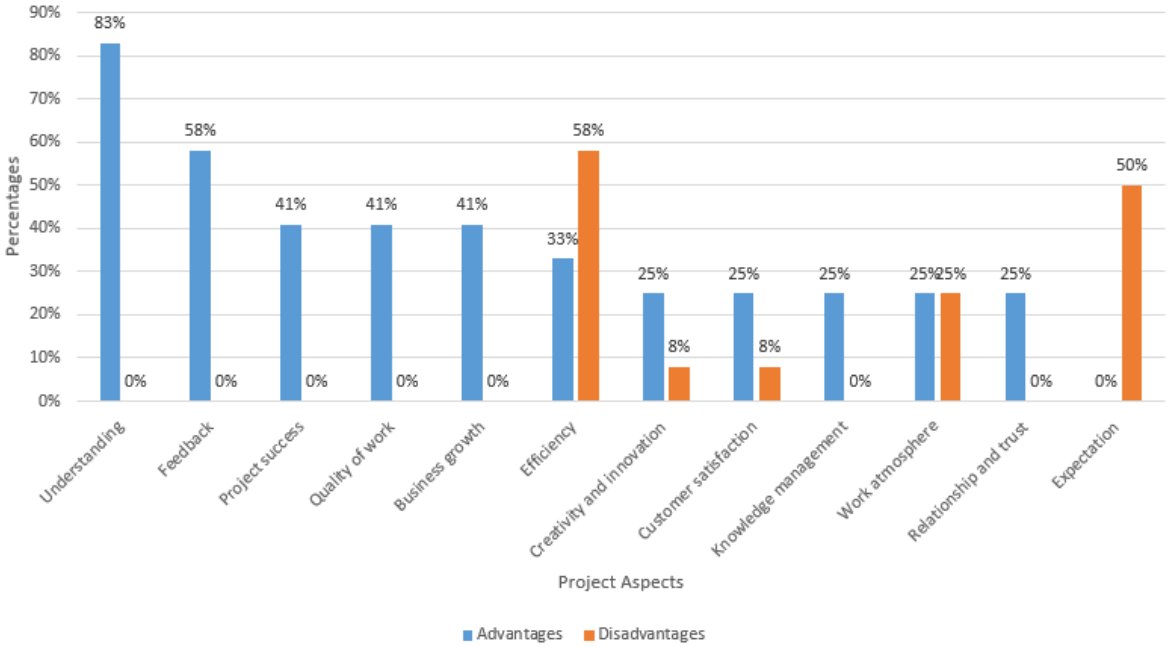

Figure 4.11 Ratio of advantages and disadvantages of customer collaboration on project aspects

Among 12 software project aspects, 11 aspects can received the positive influence of customer collaboration as shown in Table 4.6.

Table 4.6. Advantages of customer collaboration

Aspects	Description	Studies	Frequency	Percentage
Understanding	Customer collaboration can improve the understanding between customers and developers	[7], [11], [12], [18], [30], [33], [38], [40], [42], [45]	10	83%
Feedback	Customer collaboration can give quick feedback and help the discovery of issues of interface design and software bug	[7], [18], [21], [26], [27], [38], [39]	7	58%
Project success	High level that customer collaboration leads to project success	[13], [14], [22], [29], [42]	5	41%
Quality of work	Customer collaboration enhances the quality of work such as quality of requirements or the quality of the end software product	[10], [11], [21], [29], [30]	5	41%
Business growth	Customer collaboration can benefit a software organization in maintaining and improving their business by preventing the reputation loss and promoting competitive advantage	[18], [24], [33], [35], [43]	5	41%
Efficiency	Customer collaboration makes development work can be done in less effort, less time, and less cost.	[9], [28], [39], [43], [49]	5	41%
Creativity and	Customer collaboration	[43], [50], [51]	3	25%

innovation	brings creativity and innovation to software development projects			
Customer satisfaction	Customer collaboration improves customer satisfaction of end software product	[2], [10], [17]	3	25%
Knowledge management	Knowledge sharing via customer collaboration gives valuable input to the project	[31], [50], [51]	3	25%
Work atmosphere	Customer collaboration can lead to a more pleasurable work atmosphere by reducing workload and lessening the pressure of over-commitment of few individuals	[12], [18], [51]	3	25%
Relationship and trust	Customer collaboration can strengthen customers trust and therefore strengthen their relationship with the development organization	[24], [30], [42]	3	25%

Customer collaboration can create disadvantages to 5 software development project aspects as can be seen in Table 4.7.

Table 4.7. Disadvantages of customer collaboration

Aspects	Description	Studies	Frequency	Percentage
Efficiency	Customer collaboration is time consuming and slow down development work	[12], [25], [26], [29], [31], [34], [39], [44]	8	67%
Expectation	Customer collaboration brings unrealistic expectation of project roles and end product	[19], [26], [27], [34], [44]	5	50%
Work atmosphere	Customer collaboration can make the work atmosphere become unpleasant due to constant interruption	[20], [27], [34]	3	25%
Creativity and innovation	Customer collaboration does not bring creativity and innovation to a project	[47]	1	8%
Customer satisfaction	Customer collaboration does not improve customer satisfaction of software product	[17]	1	8%

The results presented in Table 4.6 and Table 4.7, show 7 project aspects that do not receive negative impact from customer collaboration at all. They are: understanding, feedback, project success, quality of work, business growth, knowledge management, and relationship and trust. Understanding improvement between customers and development team is the most discussed advantage of customer collaboration (83%). The second most investigated advantage is the increase in feedback frequency (58%).

There are 4 project aspects - efficiency, work atmosphere, creativity and innovation and customer satisfaction – that can be affected both positively, as well as negatively by customer collaboration. Within these 4 aspects, the decrease in efficiency is the most commonly discussed disadvantage of customer collaboration (58%).

There is only 1 project aspect that receives only negative effects of customer collaboration, which is expectation. The results show that 50% of the papers discussing advantages and disadvantages of customer collaboration only illustrated the conflicts in expectation with no positive facet mentioned.

Discussion

This section discusses the causes that contribute to the advantages and disadvantages of customer collaboration on the following project aspects: understanding and feedback, efficiency, and expectation. It can be seen that in general, customer collaboration could bring more advantages to software development projects than disadvantages. The results of RQ3 show that understanding improvement and quick feedback are the main advantages of customer collaboration, while the main disadvantages are the decrease in efficiency and unrealistic expectation. The advantages and disadvantages of customer collaboration are multi-faceted. The discussion is organized as follows: 1. the advantages on understanding and feedback aspects, 2. the advantages and disadvantages on the efficiency aspect, and 3. the disadvantages on the expectation aspect.

1. Understanding and Feedback Improvement

Among the advantages of customer collaboration found in the primary studies, improved understanding and quick feedback are the aspects that were discussed the most (83% and 58% respectively). No paper has documented disadvantages of collaboration upon these project aspects. From the context of each primary study, it could be seen that these advantages rely on the frequency of communication and the distance between customers and developers.

Projects in the primary studies have demonstrated the advantage of improved understanding rely on frequent communications between customers and developers. Among the primary studies, 7 out of 9 papers used agile methodologies, while the remaining two papers employed frequent communication despite being non-agile. The communication could be in a form of discussions, meetings or acceptance testing. Frequent communications allowed both parties to reconfirm mutual understanding of the objectives of a task. It provides an opportunity for early detection of errors in the design, code, and plan. Agile methodologies embed customer collaboration as one of the core values. Agile projects thus tend to have more collaborations and result in documented benefits of having frequent communications and interaction with customers. Non-agile methods were used in [7] and [30]. In the late 80's (before agile was developed), [7] mentioned understanding improvement and quick feedback as the advantages of customer collaboration. The project was developed using a process defined by AT&T, which incorporated early user involvement for requirements gathering. For study [30], the authors did not mention the software development used, but the benefit of understanding improvement while collaborating with customers occurred from the use of a prototyping technique called evolutionary prototyping. The technique advocates regular contacts between customers and developers via periodic face-to-face meetings. These studies show that, while Agile has become prominent in regulating frequent communication, this practice can occur independently from the software development method used based on the project management style. Thus, companies currently employing plan-driven or traditional software development

methods should not think that they need to totally transform into agile methodology in order to achieve such benefits.
The projects that found quick feedback to be an advantage of customer collaboration mostly used agile methodology's Extreme Programming (XP) as a software development method (43%). Frequent communication between the customers and the development team provides the mechanism for quick feedback. However, frequent communication does not promise quality of the feedback received. XP emphasizes on exercising onsite collaboration and that customers should always be available to assist developers whenever they need clarification. Thus, it is obvious that when both parties existed on a co-located space, communications can occur more easily and frequently as opposed to having a long distance communication. One primary study investigated the effects of long distance collaboration in agile and confirmed that distance can hinder frequency of collaboration due to the problems of coordinating and time zone difference [18]. Especially if the development team have never met the customers face to face before, there could be difficulties in establishing trust. This could later become the unwillingness to collaborate, and further on lessens the numbers of feedback loop.
It can be seen that by using agile methodologies, customer collaboration can easily benefit from understanding and feedback improvement. However, plan driven projects can benefit from having frequent meetings and check points with customers while agile projects can suffer from the lack of customer feedback in a long distance environment. Thus, in order to achieve these advantages, it is not only about the software development methodologies used, but also how the project members maneuver the interactions.

2. Efficiency Decrease

Customer collaboration can increase or decrease the efficiency of a software development project. The results have shown that 33% of the papers mentioned the increase of efficiency, whereas 58% claimed the decrease in efficiency. This means that the decrease of efficiency is the most discussed disadvantage of customer collaboration among the primary studies. The background context that papers discussing the increase in efficiency has in common is the use of agile methodology, while the common context for those with efficiency decrease is of small and medium sized organizations.
For those that discussed the increase in efficiency, 3 out of 4 papers mentioned using agile methodologies as a software development method. When examined further it was found that the frequent feedbacks and on-site collaboration are elements from agile that contribute to the improvement of efficiency. It is found that, 2 out of the 4 papers, are the same papers that discussed the advantage of frequent feedback [18], [39]. Via on-site collaboration, the frequent feedback from real users enabled errors or mismatches in understanding to be resolved earlier, thus more economical. It prevents developers from wasting efforts in misinterpreting requirements or developing unwanted features that later have to be discarded [18], [39]. On-site collaboration is also found to be more efficient than long distance collaboration in terms of development speed and the quality of accepted features [44].
Customer collaboration's most significant disadvantage is that it can decrease a project's efficiency. This is mostly occurred in small and medium sized organizations (33% of the paper investigated this disadvantage). The limitation of resources could be the cause of difficulties in allocating assets to accommodate frequent communications. One study mentioned that they do not have a dedicated personnel for requirement engineering tasks, but the role was responsible by the project manager. With the constraint in resources, the development team might want to prioritize on technical tasks more than collaboration tasks. For instance, [29] reported that the development team might decide not to involve customers if there is time pressure. Since small and medium companies have limited resources, they might pay more attention to how their investment can enhance the quality of work than a large

company would. Hence, resulted in a more documented dismay from smaller sized companies.
Therefore, it is vital to ensure the quality of the customers' input, especially in small and medium sized companies. The studies that mentioned the increase in efficiency seem to receive invaluable input from the customers. On the hand, the studies mentioned the decrease in inefficiency encountered matters such as, customer's input is vague or that customers also do not know how the direction of the new development project should be. Avoiding customer collaboration might bring the development tasks to finish faster, however the finished software might not be what the customer actually wanted. In order to reduce this adverse effect, small and medium sized companies should select the stakeholder for collaboration who are knowledgeable.

3. Unattainable Expectation

According to the results, customer collaboration causes only disadvantages to expectation aspect of software project development. Project members might develop unrealistic expectation during the collaboration process. There are two kinds of unrealistic expectation: an expectation of a role and an expectation of a product. The results revealed that 90% of the projects reporting these disadvantages were using agile methodology. Hence, it could be assumed that while agile manifests the unrealistic expectation project members developed during the collaboration, it also provides an opportunity to remediate the situation.
In terms of role expectation, there are the mismatches in the perception of either one's role or another party's role. The unrealistic expectation of the customer's role occurred when the customers become too absorbed in the role. For example, one user representative started to become aggressive to other users [19]. Or another example is when a group of superusers would like to have access to the code so that they could test and modify by themselves [20]. For the unrealistic expectation of another party's role, it is shown as one party expects that others should give more than they should. For example, a development team expected the customer to be able to give answers to them right away [27]. As these mentioned examples occurred in agile environment, it should be recalled that one of the objectives for having frequent communication with customers is to detect misalignments between customers and development team. Study [26] which was conducted in XP environment said that, when presenting working software to users, the users asserted that developers should provide beyond what the users want. This was the opportunity for the development team to explain scope of their role. Once explained, the project continued and end successfully according to the customers. Thus, reinforce that although unrealistic expectation in roles could occur during customer collaboration, it is also the opportunity to improve it.
There is only one empirical study on the unattainable expectation of the customers towards a product. However, the finding is unique. The author explained that excessive collaboration could cause unattainable expectation from the customers [44]. This study explained that when the customers engaged more, they formed higher expectation on the product. Even though the software is well functioned, the customers might not be satisfied if it does not meet their high expectation. This study agrees with the finding of Bano and Zowghi that collaboration or involvement should be done in moderation. It is suggested by [44] that, for new project, the moderate level of collaboration for customers should not exceed 22% of overall development time, while for maintenance project the moderate collaboration is 15% of overall development time. Similarly, Bano and Zowghi's concluded that user should involve more in a project with high uncertainties and that high level of collaboration have less or no impact on projects with low uncertainties [54]. However, while [44] advised against excessive level of collaboration as it could decrease customer's satisfaction due to their high expectation; [54] supplemented

that the more users collaborate, the more they become satisfy due to the psychological sense of control.
Therefore it is advised that project leaders should expect conflicts in expectation to happen and prepare both preventative and reactive measures for it. An appropriate level of collaboration is proposed by [44] to prevent customers from developing unattainable expectation of a product. However, the result of [44] might not be able to generalize to all software projects. Project leaders should determine the appropriate level of collaboration that best suit each project. The use of mockups or working software might be useful technique to keep the perception of all parties in line.

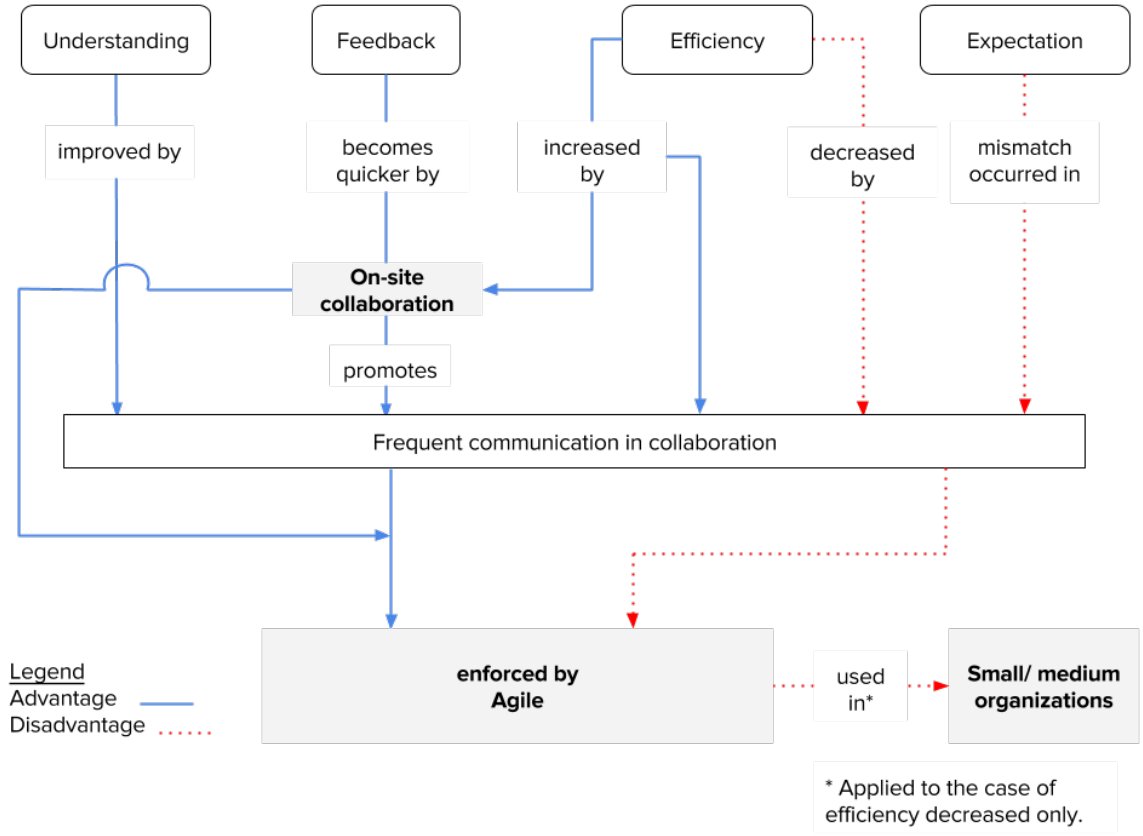

Figure 4.12 Simplified relationships of customer collaboration's advantages and disadvantages

Figure 4.12 simplified the relationship between the advantages, disadvantages, and the context of software project that caused them based on the results of this SLR. The main contexts that contribute to the advantage or disadvantage of collaboration are: software development methodology, collaboration proximity, and the size of development organization. The main variable is the frequency of communication. The software development methodology that is suitable for frequent communication collaboration is agile methodology. Close proximity collaboration provides more efficiency and quicker feedback. The increase in communication frequency can improve understanding, increase feedback loop, and increase project efficiency. However, if the communication frequency is too excessive, it could reduce the project efficiency especially in small or medium organizations and cause the customer to develop unrealistic expectation of the product.

5. Discussion

It is a common notion that customer collaboration is important for software development projects. However, approaches that enable project stakeholders to collaborate effectively are not clear. Research on this topic is fairly limited in software engineering field. This thesis investigated empirically reported challenges occurred during collaboration in software development, suggestions presented in relevant literature, and the effects of customer collaboration on different software development project aspects. Our findings suggest that the most common discussed types of challenges are stakeholders' lack of commitment, stakeholders' lack of knowledge, and geographical dispersion. The commonly reported suggestions are concerning providing effective collaboration tools, changing customers' mindset, and using user groups methods. Customer collaboration is found to be the most advantageous in the project aspects of understanding improvement and increasing feedback frequency. In the project aspect of efficiency, customer collaboration can be both advantageous and disadvantageous, where the disadvantage on efficiency is the most pervasive. Finally, the expectation project aspect surprisingly received only negative effects from customer collaboration.

This section discusses the context of the primary studies used in this SLR in relation to the related work introduced in section 2.2. This is followed by a discussion on how the important findings of each research question are related and how they correspond to the related work. Finally suggestions for future research are presented.

Context of the primary studies

This thesis provides a different perspective from the current evidence-based studies on customer collaboration in software development. The largest part of the primary studies used for this thesis is case studies and industrial reports, while Bano and Zowghi's primary studies are mostly based on surveys. Thus, the findings of this thesis are more representative of contextual industrial phenomenon than [54], [55]. We did not include secondary studies in the review as Abelien and Paech did, due to the different scopes between SLR and systematic mapping study. Bano and Zowghi [54], [55] analyzed their findings from the perspective of research methodologies. This is beneficial to their researches since the results were to be used as a starting point for the design of their case study. However, as one of the thesis' aims is to present recommendations to practitioners, we analyzed the findings based on software development project background settings.

Background settings that were found to be related to the major findings are industries of customers' organization, small and medium sized software development organization, design and requirements engineering activities, agile methodologies, and long distance collaboration. Agile software development methodology and long distance collaboration are the two major background contexts among the primary studies. This corresponded to the current trend of software development [59], [61]. However, it is unexpected to find that majority of the primary studies did not provide sufficient background setting of the studies in software development method, participants, collaboration proximity, and company size. That is, a paper that conducted in agile may not provide information on company size, while a paper that provided information on project activity, may not inform the information on the collaboration proximity. This problem was also reported by Bano and Zowghi [54]. It caused difficulty in interpreting the meaning of findings for their SLR and ours. Also, without

sufficient background information, other researchers might not be able to replicate the study and practitioners may not find it relevant to their circumstances.

Important findings

One of the main findings of this study is the challenge consisting in stakeholders' commitment towards collaboration (RQ1). This challenge is mostly found in the long distance setting. It is interesting to notice, however, that one paper proposes means of enforcing customers' commitment through changing customers' perception on the importance and benefits of the project (RQ2), as well as increasing the ease of participation [46]. This proposal is aligned with Beck's [56] suggestion to solve the challenge of uncooperative customers in XP. However, suggestions on how to increase developers' commitment are not given in the literature even though this is seen as a challenge. This agrees with Paulk's [82] observation that the challenge of securing close and effective collaboration is most likely to occur on the customer side.

Stakeholders' knowledge gap is a challenge which has to be bridged to achieve effective collaboration (RQ1). This challenge is mostly found in agile methodology settings. Barriers such as unclear terminology, cultural differences, stakeholders' reluctance in sharing their data, lack of expertise and personal communication skills could be the cause of this gap. Multiple suggestions have been proposed in this direction, such as on-site customers, developers' visits to customers, regular meetings, interviews and surveys (RQ2). However, it has been found that unrealistic customer expectations can arise from a considerable amount of collaboration which in turn could lead to low customer satisfaction (RQ3). With the involvement of more stakeholders, the knowledge flow and product quality are increased, but this tends to increase conflicts and slow down the project pace. Nevertheless, multiple studies analyzed in this thesis promote a close physical proximity collaboration between customers and developers. Therefore, when engaging a collaboration process, practitioners need to choose the right techniques and a suitable level of collaboration in order to promote knowledge transfer and at the same time ensure client satisfaction.

Customer input and feedback is regarded as highly important throughout the development process (RQ2). Different means of achieving it are proposed as general suggestions: user group techniques, demos and prototypes; or in the case of off-the-shelf software, specific customer studies are suggested (RQ2). In terms of specific suggestions, methods and techniques such as SRP, customer-specific teams and story masters were proposed. Compared to Bano and Zowghi's methods and techniques findings [55], most of them are already well-established such as participatory design, user-centered design, or agile. However, methods and techniques listed in our SLR are meant to be used as complementary with agile methodologies.

Most general suggestions on the use of effective communication tools are concerned with a long distance collaboration context (RQ2). Synchronous collaboration tools are highly needed in order to secure timely communication between the customers and developers. Tools suggestions are found to be web-based tools. These tools are useful in long distance collaboration because software artifacts can be globally shared and accessed. However, they do not provide the synchronous characteristics for instantaneous communication.

Customer collaboration can be advantageous in many software development project aspects (RQ3). The most discussed project aspects that benefit from customer collaboration are understanding and feedback. Other aspects such as work quality, efficiency, and creativity and innovation were also shown to benefit from customer collaboration. Each study investigates the advantages or disadvantages of customer collaboration from different levels of project aspects. For example, understanding and satisfaction are in the inner-layer aspects that can contribute to an outer-layer of project aspects such as business growth or project

success. Findings from Abelein and Paech [52] and Bano and Zowghi [54] indicated that, in general customer collaboration or involvement could lead to system success. In this thesis, such findings are one of the advantages customer collaboration could bring to software projects. Nevertheless, this thesis have found that system success was not the most popular project aspect that were investigated (RQ3), but was ranked the third.

One noteworthy finding is, communication frequency appears to be the main factor that influences the advantages or disadvantages of the project aspects of understanding, feedback, efficiency, and expectation. Increasing communication frequency can lead to advantages in project aspects such as understanding, feedback and efficiency (RQ3). On the other hand, excessive communication can lead to the disadvantages in the aspects of efficiency and expectation. Interestingly, frequent communication can lead to more disadvantages in project efficiency as opposed to increase advantages. Many papers discussing these advantages and disadvantages were conducted in agile settings. The relation between the advantages and the agile development methodologies appears to correspond to the agile value of promoting communication and collaboration between customers and developers. Notwithstanding, there are conflicting recommendations from agile methodologists on the appropriate level of communication frequency during a software development project. Schwaber and Sutherland [87] did not mention this concern in The Scrum Guide at all. Beck's XP theoretically requires full-time availability from the customers [56]. Cockburn's [62] Crystal methods, despite not being used in any of the primary studies, mentioned that a weekly access to an expert user is valuable.

It is surprising to find that the project aspect of expectation only receives negative effects from customer collaboration. This disadvantage was reported to occur in agile development setting. One would assume that frequent communication in agile could result in a more realistic expectation. Although, this might be due to the small number of primary studies, the concept of expectation is of subjective nature. The aspect of expectation seems to intertwine with the aspects of understanding. However, primary studies that reported the advantage of understanding improvement do not overlap with those studies that reported the disadvantage of unrealistic expectation. Still, the primary studies that reported the disadvantages of unrealistic expectation emphasize on the "anticipation" while studies that mentioned understanding emphasize on the knowledge of a task. When analyzing the primary studies, it is found that some of the judgment of whether collaboration has benefits or drawbacks for an aspect may be subjective to the authors of the respective study. Kautz's [26] study collaboration can be a disadvantage because it causes conflicts in expectation between customers and developers during an iteration, but it could also be interpreted as an advantage. This is because the expectation could be aligned in the next iteration, as opposed to leaving the mismatch to remain until later stage of the project. Bano and Zowghi [54] suggested that the conflicting results of the studies in user involvement could occur from the ambiguity of terminology and different context in the software project. Our findings confirm their statement and also expand it with the element of an author's subjectivity.

Recommendations for practitioners

The implication of the findings lead to an overview that can be used as recommendations for practitioners. However, as each software project may have its own constraints, an in-depth research on how to collaborate effectively on each constraint is still needed.

1. Selection of representatives

Project leaders should carefully select development team representatives. If the project leaders have the authority to select customer representative, it should also be done carefully.

a. It's not necessary for everyone in the development team members to interact with the customer representative on a day to day basis. For requirements engineering activities, one requirement engineer should be the only link between customers and development team [16]
b. The customer representatives should have technical background knowledge. He or she should have a clear understanding of the requirements and project goals. Customer representatives needs to immediately communicate any possible changes that are being discussed by the customer's team. Providing knowledge to customers might assist them in better problems description [36], [38].
c. The customers should be able to allocate most of their time to the project and needs to be available immediately if questions or clarification is required by the project team.

2. Long distance collaboration
Both customer and project management will benefit from an investment in high quality collaboration tools. Select the tools that provides adequate after-sale support and provide sufficient training of the tools to developers [31].

3. Long distance collaboration – global collaboration
a. Follow the recommendation 2
b. It is important to learn about the culture of project individual members and adjust accordingly [9], [31].
c. Meeting facilitators should be considerate of each member's time difference.

4. Trust establishment
a. Facilitate a face-to-face meeting at least once. A face-to-face meeting can develop the trust between the customers and the development team. If a face-to-face meeting is not appropriate or possible, consider a video conference session where each member can be introduced.
b. In all cases management needs to increase informal communication and reduce any conflicts between the customers and the development team [9], [18].

5. Enhancing commitment
a. If the cause of insufficient commitment is due to customers' workload, a story master role [41] may lessen the workload of a customer, leading to the ability to have more commitment on important tasks.
b. Change customers' mindset by communicating the benefit of the end results of the project.
c. Top-down organization level support may be needed during a larger project that involves customers from different companies.

6. Collaboration frequency
a. Project leaders should find the appropriate frequency of collaboration. If the customer representatives are not a dedicated member, too frequent communication with the project team employing agile methodology may become a distraction from their daily routine. This could create annoyance and work-priority conflict, as well as lead to frustration for the project team which needs immediate collaborative communication.
b. It is suggested by [44] that for new projects the moderate frequency of collaboration for customers should not exceed 22% of overall development time.
c. For a maintenance project the moderate collaboration is 15% of overall development time.

7. Continuous improvement
Continuously evaluating the quality of collaboration, especially in long term projects such as in [48] should be beneficial. Also, customer collaboration should not only be limited to new

product development. Extending the collaboration to after-launched phase can be speculated to benefit customer satisfaction and maintain software quality.

By conducting this SLR, we have listed challenges, proposed suggestions and tools, and presented project aspects that can be affected by customer collaboration. The findings are intricately connected. Considering various background of software projects, it is difficult to produce one universal rule to achieve effective customer collaboration. However, understanding the results of each RQ should assist practitioners in weighing the advantages and disadvantages proposed by the literature. At the same time, practitioners can attempt to investigate challenges of their organization and means of applying the suitable suggestions in order to overcome them. Due to small numbers of relevant publications, along with incomplete information given by the authors of each studies, the generalizability of this SLR is minimum. Still, we have produced results that reflect what have been empirically studied with a focus on the industrial context.

Further research

The findings in this thesis also revealed the gap in the current state of the primary studies. No one single suggestion can provide a solution to all the projects, as different project has its own objectives and constraints. Therefore, the future studies should be towards a specific solution for a specific challenges, such as geographical dispersion or smaller sized companies. The following are topics for further research that should be considered.

- Finding concrete means of changing software project stakeholders' attitude to become more committed to collaboration.
- Investigate the topic of managing customer expectation during collaboration in software development projects.
- Overcoming the challenge of customers' work domain understanding from a global development perspective.
- Finding the rationale of choice of methods, tools and processes for effective collaboration in a global development perspective.
- Investigating possible collaboration tools to be used by practitioners aiming for synchronous communication, especially in a global development project.
- Investigating the right amount of communication in collaboration to maximize the project's benefits, especially in the aspect of efficiency.
- Investigating cases when a smaller development company collaborates with a different sizes customer companies. The hypothesis could be larger companies tend to demand more requirements change as they are they are more superior in terms of finance and reputation.
- Establishing a set of terminologies that can reduce ambiguity when researchers conduct studies on a more subjective topic such as collaboration or involvements with customers.

6 Conclusion

This thesis presents findings from a systematic literature review on challenges, suggestions of processes and tools, and advantages along with disadvantages in customer collaboration during software development.

Many of the reviewed studies did not provide sufficient information on the context of the studies, nor did they mention limitations or threats to validity. This created difficulties for the analysis of evidence-based secondary studies and for other researchers to replicate the studies. Moreover, it is not useful for practitioners if they cannot apply the studies to their project context.

The most challenging aspects of customer collaboration are found to be the commitment of stakeholders together with the customers' unfamiliarity with the software development field, the developers' misunderstanding of customers' domain knowledge, and geographical dispersion which leads to ineffective communication.

A large number of the studied publications propose tools, as well as methods and processes. The majority of proposed tools are web-based applications meant for a better design or requirements elicitation process. The methods and processes are mostly connected to iterative development and suggest physical proximity with the customers during the development process. A larger part of the studied literature provides general suggestions such as the implementation of communication tools which allow for immediate and effective response, the use of user group techniques, iterative development with the usage of customer feedback throughout the development process, and a change in the customers' mindset towards collaboration.

Most advantages of customer collaboration lie in the area of enhancing understandability and frequent feedback. Customer collaboration can also improve project efficiency. However, too frequent communication could result in the disadvantage of project inefficiency, which is the most commonly found drawback of customer collaboration. Another major disadvantage of customer collaboration is it can lead to unreasonable expectation.

In general there are more reported evidence on advantages in customer collaboration than disadvantage. In order to maximize the benefit from it, practitioners need to analyze the challenges that each project is facing and select an appropriate solution to overcome them. This thesis provides an outline recommendations for practitioners on how to tackle some of the most commonly found challenges found via the SLR. As the results are based on a small number of publications, extending the search terms and data sources should provide more generalizable results. One of the directions of future researches should be to study how to collaborate with customers in a specific context, such as in a global software development environment. This thesis broke down the challenges and solutions of customer collaboration that both researchers and practitioners can further examine.

7. Acknowledgements

This master thesis could not have been finished without the help and support from Jeppesen Systems AB's personnel, Chalmers' professors and our families. It is our great pleasure to acknowledge people who have given us guidance, help and encouragement.

We would like to firstly thank Tomat Klemets from Jeppesen for kindly giving us opportunities for this thesis. Also, thanks to David Karlsson and Viktor Forsman for offering their advice and assistance throughout this project. Furthermore thanks to all the rest of Jeppesen personnel who provided us with useful insights and opinions.

Secondly, thanks to Dr. Ana Magazinius for her constant guidance, personal attention, suggestions and endless encouragement and full support during last eight months of our master program.

Special thanks go to our examiner Dr. Richard Berntsson Svensson for his important suggestions and remarks in this research and for his time and effort in service of this master thesis.

Finally, we would express my gratitude to our parents, whose support and constant encouragement helped us through the hard times of this research.

8. References

Systematic review references

[1] R. Andersen and A. I. Mørch, "Get satisfaction: Customer engagement in collaborative software development," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 7897 LNCS, ed, 2013, pp. 235-240.

[2] Z. Bakalova and M. Daneva, "A comparative case study on clients participation in a 'traditional' and in an Agile software company," presented at the Proceedings of the 12th International Conference on Product Focused Software Development and Process Improvement, Torre Canne, Brindisi, Italy, 2011.

[3] P. G. Beerthuizen, "Design decomposition strategies, and their impact on requirements elicitation and specification," in *European Space Agency, (Special Publication) ESA SP*, 2006.

[4] B. Begier, "Users' involvement may help respect social and ethical values and improve software quality," *Information Systems Frontiers,* vol. 12, pp. 389-397, 2010.

[5] G. Bjerknes and L. Mathiassen, "Improving the customer-supplier relation in IT development," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, 2000, p. 10 pp.

[6] E. Brandt, "Action research in user-centred product development," *AI & SOCIETY,* vol. 18, pp. 113-133, 2004/04/01 2004.

[7] H. S. Burns and R. A. Halliburton, "Tackling productivity and quality through customer involvement and software technology," in *Global Telecommunications Conference and Exhibition 'Communications Technology for the 1990s and Beyond' (GLOBECOM), 1989. IEEE*, 1989, pp. 631-635 vol.1.

[8] S. Cohen, D. Dori, and U. de Haan, "A software system development life cycle model for improved stakeholders' communication and collaboration," *International Journal of Computers, Communications and Control,* vol. 5, pp. 20-41, 2010.

[9] D. E. Damian and D. Zowghi, "RE challenges in multi-site software development organizations," *Requirements Engineering,* vol. 8, pp. 149-60, / 2003.

[10] V. V. Das, "Involvement of Users in Software Requirement Engineering," in *Information Technology, (ICIT 2007). 10th International Conference on*, 2007, pp. 230-233.

[11] I. Erfurth and W. Rossak, "DigiShare - A participative methodology to build linked user interfaces," in *Proceedings of the 3rd IASTED International Conference on Human-Computer Interaction, HCI 2008*, 2008, pp. 261-266.

[12] T. E. Faegri and G. K. Hanssen, "Collaboration, Process Control, and Fragility in Evolutionary Product Development," *Software, IEEE,* vol. 24, pp. 96-104, 2007.

[13] G. G. Gable, "Consultant engagement for computer system selection. A pro-active client role in small businesses," *Information and Management,* vol. 20, pp. 83-93, 02/ 1991.

[14] S. N. Geethalakshmi and A. Shanmugam, "Success and failure of software development: practitioners' perspective," in *International MultiConference of Engineers and Computer Scientists 2008, 19-21 March 2008*, Kwun Tong, China, 2008, pp. 915-21.

[15] M. Gomez and J. Cervantes, "User interface transition diagrams for customer-developer communication improvement in software development projects," *Journal of Systems and Software,* vol. 86, pp. 2394-410, 2013.

[16] C. Hansson, Y. Dittrich, and D. Randall, "Agile processes enhancing user participation for small providers of off-the-shelf software," in *Extreme Programming and Agile Processes in Software Engineering. 5th International Conference, XP 2004. Proceedings, 6-10 June 2004*, Berlin, Germany, 2004, pp. 175-83.

[17] S. Heidenreich, K. Wittkowski, M. Handrich, and T. Falk, "The dark side of customer co-creation: exploring the consequences of failed co-created services," *Journal of the Academy of Marketing Science,* pp. 1-18, 2014/05/14 2014.

[18] R. Hoda, J. Noble, and S. Marshall, "The impact of inadequate customer collaboration on self-organizing Agile teams," *Information and Software Technology,* vol. 53, pp. 521-534, 2011.

[19] H. Holmstrom and O. Henfridsson, "Customer role ambiguity in community management," in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences, 7-10 Jan. 2002*, Los Alamitos, CA, USA, 2002, pp. 2190-7.

[20] H. Holmstrom and O. Henfridsson, "Improving packaged software through online community knowledge," *Scandinavian Journal of Information Systems,* vol. 18, pp. 3-36, / 2006.

[21] K. Holtzblatt, "Customer-centered design for mobile applications," *Personal and Ubiquitous Computing,* vol. 9, pp. 227-237, 2005/07/01 2005.

[22] V. Holzmann and I. Panizel, "Communications management in scrum projects," in *7th European Conference on Information Management and Evaluation, ECIME 2013*, 2013, pp. 67-74.

[23] I. Inayat, M. A. Noor, and Z. Inayat, "Successful product-based agile software development without onsite customer an industrial case study," *International Journal of Software Engineering and its Applications,* vol. 6, pp. 1-14, 2012.

[24] J. Kabbedijk, S. Brinkkemper, S. Jansen, and B. van der Veldt, "Customer Involvement in Requirements Management: Lessons from Mass Market Software Development," in *Requirements Engineering Conference, 2009. RE '09. 17th IEEE International*, 2009, pp. 281-286.

[25] D. Kane, M. Hohman, E. Cerami, M. McCormick, K. Kuhlmman, and J. Byrd, "Agile methods in biomedical software development: a multi-site experience report," *BMC Bioinformatics,* vol. 7, pp. 1-12, 2006/05/30 2006.

[26] K. Kautz, "Participatory design activities and agile software development," in *IFIP Advances in Information and Communication Technology* vol. 318, ed, 2010, pp. 303-316.

[27] J. Koskela and P. Abrahamsson, "On-Site customer in an XP Project: Empirical results from a case study," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 3281, ed, 2004, pp. 1-11.

[28] B. Krogstie and B. Bygstad, "Cross-community collaboration and learning in customer-driven software engineering student projects," in *Software Engineering Education & Training, 2007. CSEET'07. 20th Conference on*, 2007, pp. 336-343.

[29] S. Kujala, M. Kauppinen, L. Lehtola, and T. Kojo, "The role of user involvement in requirements quality and project success," in *Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on*, 2005, pp. 75-84.

[30] M. Lang and J. Duggan, "A Tool to Support Collaborative Software Requirements Management," *Requirements Engineering,* vol. 6, pp. 161-172, 2001/10/01 2001.

[31] K. Liukkunen, K. Lindberg, J. Hyysalo, and J. Markkula, "Supporting Collaboration in the Geographically Distributed Work with Communication Tools in the Remote District SME's," in *Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on*, 2010, pp. 155-164.

[32] S. Martikainen, M. Korpela, and T. Tiihonen, "User participation in healthcare IT development: A developers' viewpoint in Finland," *International Journal of Medical Informatics,* vol. 83, pp. 189-200, 2014.

[33] L. Miller, "Case study of customer input for a successful product," in *Agile Conference, 2005. Proceedings*, 2005, pp. 225-234.

[34] S. Mitra and T. A. Bullinger, "Using formal software development methodologies in a real-world student project: an experience report," *Journal of Computing Sciences in Colleges,* vol. 22, pp. 100-108, 2007.

[35] K. Molokken-Ostvold and K. M. Furulund, "The Relationship between Customer Collaboration and Software Project Overruns," in *Agile Conference (AGILE), 2007*, 2007, pp. 72-83.

[36] P. Noraini Che and A. M. Zin, "Requirement Elicitation: Identifying the Communication Challenges between Developer and Customer," *International Journal of New Computer Architectures and their Applications,* vol. 1, pp. 371-83, / 2011.

[37] H. H. Olsson, J. Bosch, and H. Alahyari, "Customer-Specific Teams for Agile Evolution of Large-Scale Embedded Systems," in *Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on*, 2013, pp. 82-89.

[38] M. Pichler, H. Rumetshofer, and W. Wahler, "Agile requirements engineering for a social insurance for occupational risks organization: a case study," in *14th IEEE International Requirements Engineering Conference, 11-15 Sept. 2006*, Los Alamitos, CA, USA, 2006, p. 6 pp.

[39] C. Pieterson and A. Cousineau, "An involvement continuum for custom software development: the `CATS' example," in *HUSITA-2. Human Service Information Technology Applications. 1991 International Conference, 26-30 June 1991*, USA, 1993, pp. 179-86.

[40] M. Pikkarainen, J. Haikara, O. Salo, P. Abrahamsson, and J. Still, "The impact of agile practices on communication in software development," *Empirical Software Engineering,* vol. 13, pp. 303-37, 06/ 2008.

[41] A. Read, N. Arreola, and R. O. Briggs, "The Role of the Story Master: A Case Study of the Cognitive Load of Story Management Tasks," in *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, 2011, pp. 1-10.

[42] J. Segal and C. Morris, "Scientific End-user Developers and Barriers to User/Customer Engagement," *Journal of Organizational and End User Computing,* vol. 23, pp. 51-63, 10/ 2011.

[43] M. Shammi, S. Overbeek, R. Verburg, M. Janssen, and Y. H. Tan, "Agile process for integrated service delivery," in *CEUR Workshop Proceedings*, 2011, pp. 31-40.

[44] R. Subramanyam, F. L. Weisstein, and M. S. Krishnan, "User participation in software development projects," *Commun. ACM,* vol. 53, pp. 137-141, 2010.

[45] N. Surendra, "Using an ethnographic process to conduct requirements analysis for agile systems development," *Information Technology and Management,* vol. 9, pp. 55-69, 2008/03/01 2008.

[46] R. Thakurta and R. Roy, "Determinants of User Involvement in Software Projects," in *2012 45th Hawaii International Conference on System Sciences (HICSS), 4-7 Jan. 2012*, Los Alamitos, CA, USA, 2012, pp. 4894-903.

[47] A. Weterings and R. Boschma, "Does spatial proximity to customers matter for innovative performance?. Evidence from the Dutch software sector," *Research Policy,* vol. 38, pp. 746-755, 2009.

[48] A. M. Wichansky, "Customer boards as vehicles of change in enterprise software user experience," in *Human Interface and the Management of Information. Designing Information Environments. Symposium on Human Interface 2009, 19-24 July 2009*, Berlin, Germany, 2009, pp. 316-22.

[49] A. Wojciechowski, M. Wesolowski, and W. Complak, "Experimental evaluation of 'on-site customer' XP practice on quality of software and team effectiveness," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 6428 LNCS, ed, 2010, pp. 269-278.

[50] H. Wu and C. Lan, "Community Collaboration for ERP Implementation," *Software, IEEE,* vol. 26, pp. 48-55, 2009.

[51] W. Zhang, L. Yi, H. Zhao, and Z. Jin, "Feature-oriented stigmergy-based collaborative requirements modeling: an exploratory approach for requirements elicitation and evolution based on web-enabled collective intelligence," *Science China Information Sciences,* vol. 56, pp. 1-18, 2013/08/01 2013.

References

[52] U. Abelein and B. Paech, "Understanding the Influence of User Participation and Involvement on System Success – a Systematic Mapping Study," *Empirical Software Engineering,* pp. 1-54, 2013/12/05 2013.

[53] C. M. Axtell, P. E. Waterson, and C. W. Clegg, "Problems integrating user participation into software development," *International Journal of Human-Computer Studies,* vol. 47, pp. 323-345, 1997.

[54] M. Bano and D. Zowghi, "User involvement in software development and system success: a systematic literature review," in *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*, 2013, pp. 125-130.

[55] M. Bano and D. Zowghi, "Users' involvement in requirements engineering and system success," in *Empirical Requirements Engineering (EmpiRE), 2013 IEEE Third International Workshop on*, 2013, pp. 24-31.

[56] K. Beck, "Embracing change with extreme programming," *Computer,* vol. 32, pp. 70-77, 1999.

[57] B. Boehm, "Get ready for agile methods, with care," *Computer,* vol. 35, pp. 64-69, 2002.

[58] B. Boehm and R. Turner, "People factors in software management: lessons from comparing agile and plan-driven methods," *CrossTalk: The Journal of Defense Software Engineering,* vol. 16, pp. 4-8, 2003.

[59] J. Bosch and P. Bosch-Sijtsema, "From integration to composition: On the impact of software product lines, global development and ecosystems," *Journal of Systems and Software,* vol. 83, pp. 67-76, 2010.

[60] M. Bruce, F. Leverick, D. Littler, and D. Wilson, "Success factors for collaborative product development: a study of suppliers of information and communication technology," *R&D Management,* vol. 25, pp. 33-44, 1995.

[61] E. Carmel and R. Agarwal, "The maturation of offshore sourcing of information technology work," in *Information systems outsourcing*, ed: Springer, 2006, pp. 631-650.

[62] A. Cockburn, *Crystal clear: a human-powered methodology for small teams*: Pearson Education, 2004.

[63] A. Cockburn, *Agile software development: the cooperative game*: Pearson Education, 2006.

[64] A. Cockburn and J. Highsmith, "Agile software development: The people factor," *Computer,* vol. 34, pp. 131-133, 2001.

[65] J. Cohen, "A Coefficient of Agreement for Nominal Scales," *Educational and Psychological Measurement,* vol. 20, pp. 37-46, 1960.

[66] M. Coram and S. Bohner, "The impact of agile methods on software project management," in *Engineering of Computer-Based Systems, 2005. ECBS'05. 12th IEEE International Conference and Workshops on the*, 2005, pp. 363-370.

[67] A. M. Davis, E. H. Bersoff, and E. R. Comer, "A strategy for comparing alternative software development life cycle models," *Software Engineering, IEEE Transactions on,* vol. 14, pp. 1453-1461, 1988.

[68] M. Fowler and J. Highsmith, "The agile manifesto," *Software Development,* vol. 9, pp. 28-35, 2001.

[69] G. Gladden, "Stop the life-cycle, I want to get off," *ACM SIGSOFT Software Engineering Notes,* vol. 7, pp. 35-39, 1982.

[70] S. Green, "Cochrane handbook for systematic reviews of interventions version 5.1. 0 [updated March 2011]," *The Cochrane Collaboration,* 2011.

[71] J. Grudin, "Obstacles to user involvement in software product development, with implications for CSCW," *International Journal of Man-Machine Studies,* vol. 34, pp. 435-452, 1991.

[72] J. D. Herbsleb and D. Moitra, "Global software development," *Software, IEEE,* vol. 18, pp. 16-20, 2001.

[73] J. Iivari, H. Isomäki, and S. Pekkola, "The user–the great unknown of systems development: reasons, forms, challenges, experiences and intellectual contributions of user involvement," *Information Systems Journal,* vol. 20, pp. 109-117, 2010.

[74] M. Jorgensen and M. Shepperd, "A systematic review of software development cost estimation studies," *Software Engineering, IEEE Transactions on,* vol. 33, pp. 33-53, 2007.

[75] M. Keil and E. Carmel, "Customer-developer links in software development," *Communications of the ACM,* vol. 38, pp. 33-44, 1995.

[76] B. A. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," 2007.

[77] E. Kommission, *The new SME definition: User guide and model declaration*: European Comm., Publication Office, 2005.

[78] R. E. Kraut and L. A. Streeter, "Coordination in software development," *Communications of the ACM,* vol. 38, pp. 69-81, 1995.

[79] S. Kujala, "User involvement: a review of the benefits and challenges," *Behaviour & information technology,* vol. 22, pp. 1-16, 2003.

[80] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *biometrics,* pp. 159-174, 1977.

[81] S. Nerur, R. Mahapatra, and G. Mangalaraj, "Challenges of migrating to agile methodologies," *Communications of the ACM,* vol. 48, pp. 72-78, 2005.

[82] M. C. Paulk, "Agile methodologies and process discipline," *Institute for Software Research,* p. 3, 2002.

[83] M. Poppendieck and T. Poppendieck, *Lean software development: an agile toolkit*: Addison-Wesley Professional, 2003.

[84] D. Radjenović, M. Heričko, R. Torkar, and A. Živkovič, "Software fault prediction metrics: A systematic literature review," *Information and Software Technology,* vol. 55, pp. 1397-1418, 2013.

[85] N. Ramasubbu and R. K. Balan, "Globally distributed software development project performance: an empirical analysis," in *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*, 2007, pp. 125-134.

[86] W. W. Royce, "Managing the development of large software systems," in *proceedings of IEEE WESCON*, 1970.

[87] K. Schwaber and J. Sutherland, "The scrum guide," *Scrum. org, October,* 2011.

[88] F. P. Seth, E. Mustonen-Ollila, O. Taipale, and K. Smolander, "Software Quality Construction: Empirical Study on the Role of Requirements, Stakeholders and Resources," in *Software Engineering Conference (APSEC), 2012 19th Asia-Pacific*, 2012, pp. 17-26.

[89] P. Tait and I. Vessey, "The Effect of User Involvement on System Success: A Contingency Approach," *MIS Quarterly,* vol. 12, pp. 91-108, 1988.

[90] R. Williams, "Managing the development of reliable software," *ACM SIGPLAN Notices,* vol. 10, pp. 3-8, 1975.

Appendix A

Data extraction form

General questions:

1. Reviewer's name
2. Reviewed date
3. Publication title
4. Author(s)

Data needed for overview study context:

1. Publication year
2. Publication type
   Mark only one oval.
   - Journal article
   - Conference proceeding
3. Methodology
   Select all that apply.
   □ Qualitative
   □ Quantitative
   □ Mixed between qualitative and quantitative
   □ Case study
   □ Action research
   □ Survey research
   □ Ground theory
   □ Not stated
   □ Other:
4. Data collection method
   Select all that apply.
   □ Questionnaires
   □ Interview: structured
   □ Interview: semi-structured
   □ Interview: unstructured
   □ Observations
   □ Not stated
   □ Other:
5. Industry
   Select all that apply.
   □ Automotive
   □ Software
   □ Telecommunication
   □ Medical
   □ Space
   □ Not stated
   □ Other:
6. Size of the company
   Select all that apply.
   If not categorized in the paper, refer to EU's definition [77]

http://ec.europa.eu/enterprise/policies/sme/facts-figures-analysis/sme-definition/index_en.htm
□ Small
□ Medium
□ Large
□ Not stated
□ Other:

7. Time frame of the study
8. Participants
   Select all that apply.
   □ Project manager
   □ Line manager
   □ Developer
   □ Tester
   □ Customer
   □ User
   □ Scrum master
   □ Product owner
   □ Not stated
   □ Other:
9. Software development process
   Select all that apply.
   □ Agile in general
   □ Agile: Scrum
   □ Agile: XP
   □ Agile: Kanban
   □ Lean
   □ Plan-driven
   □ Not stated
   □ Other:
10. Location
    Select all that apply.
    □ Long distance/ distributed
    □ On-site
    □ Not stated
    □ Other:
11. Activities
    Select all that apply.
    □ Requirements engineering
    □ Testing
    □ Implementation
    □ Design
    □ Not stated
    □ Other:

Questions for quality assessment:
This section the answer is given as No, Yes, or Other, where Other will be treated as Partial and comments.

QA1. Objectives or research goals are clearly stated
No

Yes
Other

QA2. Research questions and/or hypotheses are clearly stated
No
Yes
Other

QA3. Related work/background/literature review described
No
Yes
Other

QA4. Threats to validity or limitations identified
No
Yes
Other

QA5. Was the sample size justified? (Kitchenham)
No
Yes
Other

QA6. How clear and coherent is the reporting? (Kitchenham)
No
Yes
Other

Questions for answering research questions:

1. Proposed tools, framework, process
2. Detail of what proposed
3. Challenges of customer collaboration
4. Advantages of customer collaboration
5. Disadvantages of customer collaboration

The form ends with a comment section where the reviewer could write down any comments.

Appendix B

Table B.a. Tools supporting customer collaboration

Tools	Papers
Web-based application for prototyping shared between customers and developers	[11]
User interface transition diagram (UITD) - tool for requirements gathering. In the UITD a User Interface presentation is modelled as a state, and the events that cause transitions between the states are modelled as the transitions that lead towards another (or the same) UI presentation. In the generated graph, each vertex is the specific presentation of a UI and the outgoing edges are all the actions that the user can carry out in each UI presentation.	[15]
A prototype of a web-based knowledge repository system that supports ERP package implementations called Epics. Epics includes the following features that enable collaboration 1. Wiki 2. Social 3. Recommender and collaborative-filter systems With Epics, user can search and reuse the knowledge and contribute knowledge to design and implement an ERP by themselves. Also, software vendors can use Epics to support customer and consultants can use Epics to collaborate with customers. The tools benefit an organization by help reducing over-dependence on individual consultants while improving implementation quality and reduce risks. It can also quickly respond to urgent customers need to customize the software.	[50]
A social media platform such as Get Satisfaction allows users to suggest features and more technical users can suggest the implementation solutions as well. This type of collaborative software development is considered to be a type of "distributed EUD." Features of such social media collaboration tool include forum, good point, mood and likes.	[1]
The key features of RM-Tool are that it: - maintains a shared data dictionary within a centralised, multi-user, multimedia database with full Webbased access; - maintains requirement descriptions using both technical and non-technical techniques, with hyperlinks between corresponding descriptions; provides basic support for standard modelling notations; - supports the storage and manipulation of rich text and compound multimedia documents, which is especially useful for describing complex requirements and for demonstrating simulated implementations; - supports time- and location-independent electronic communication between a team of stakeholders using secure, authenticated connections; - supports backwards and forwards tracing of requirements, from source through to implementation; - facilitates project management, by allocating priorities, responsibilities and deadlines; - has extensive reporting capabilities, and can generate a hard-copy SRS.	[30]

Table B.b. Methods and processes supporting customer collaboration

Methods and processes	Papers
Web-based application for prototyping shared between customers and developers	[11]
User interface transition diagram (UITD) - tool for requirements gathering. In the UITD a User Interface presentation is modelled as a state, and the events that cause transitions between the states are modelled as the transitions that lead towards another (or the same) UI presentation. In the generated graph, each vertex is the specific presentation of a UI and the outgoing edges are all the actions that the user can carry out in each UI presentation.	[15]
A prototype of a web-based knowledge repository system that supports ERP package implementations called Epics. Epics includes the following features that enable collaboration 1. Wiki 2. Social 3. Recommender and collaborative-filter systems With Epics, user can search and reuse the knowledge and contribute knowledge to design and implement an ERP by themselves. Also, software vendors can use Epics to support customer and consultants can use Epics to collaborate with customers. The tools benefit an organization by help reducing over-dependence on individual consultants while improving implementation quality and reduce risks. It can also quickly respond to urgent customers need to customize the software.	[50]
A social media platform such as Get Satisfaction allows users to suggest features and more technical users can suggest the implementation solutions as well. This type of collaborative software development is considered to be a type of "distributed EUD." Features of such social media collaboration tool include forum, good point, mood and likes.	[1]
The key features of RM-Tool are that it: - maintains a shared data dictionary within a centralized, multi-user, multimedia database with full web-based access; - maintains requirement descriptions using both technical and non-technical techniques, with hyperlinks between corresponding descriptions; provides basic support for standard modelling notations; - supports the storage and manipulation of rich text and compound multimedia documents, which is especially useful for describing complex requirements and for demonstrating simulated implementations; - supports time- and location-independent electronic communication between a team of stakeholders using secure, authenticated connections; - supports backwards and forwards tracing of requirements, from source through to implementation; - facilitates project management, by allocating priorities, responsibilities and deadlines; - has extensive reporting capabilities, and can generate a hard-copy SRS.	[30]
Put in place customer specific teams which work with one selected highly	[37]

prioritized client	
Name a story master responsible for: 1. Capturing stories in meetings 2. Managing the backlog 3. Coordinating development of stories with UI designer and Development team 4. Creating and running acceptance tests The story master role is most suitable with projects with complex requirements, otherwise there might not be enough work for this role.	[41]
Models that address the customer-supplier relation in IT development. 1. SA-CMM has a strong orientation towards competition, control, and engineering. 2. SPIRE takes more balanced approach.	[5]
10-15 board members who were required to: sign a Customer Participation Confidentiality Agreement (CPCA), - attend three working meetings per year, - perform necessary pre- and post-meeting assignments (typically collecting data within their organizations), - contribute actively during working meetings and online conferences, - communicate with and support Oracle usability executives and senior staff driving action items, - facilitate Oracle access to users in their organizations for studies that would advance the goals of the group Organized meetings became a series of dialogues between Oracle and customers and among customers themselves. Oracle senior management facilitated the groups. Oracle scribes, often designers and usability engineers, took notes during working group sessions and prepared minutes and action items to be addressed between meetings. Outcomes of these groups were report formats for usability issues, Web conference training on technical topics, and sharing of opinions about the importance of Web 2.0 features.	[48]
In order to manage and enhanced customer role, Daydream introduced the concept of "Clusterball ambassadors" who were responsible for: - introduce the game to their local network of contacts such as friends, people in their hometown and people in surrounding areas. - helpful in the work of improving and realizing community activities such as administration and hosting of Clusterball tournaments - strengthen the relationship between Daydream and the rest of the gaming community - contribute both to daily company activities as well as in representing Daydream in its contact with potential customers.	[19]
The author proposed that his altered Contextual Design (CD) framework can solve problems for his specific case in building a mobile game applications. In this technique, the author meet with a group of potential users and interview/ observe their preference and opinion. The card prototype presented to the users were made as realistic as possible so that realistic feedback can be received. In the end the author presented a	[21]

working application for testing live. - Interview in real usage settings - Introduce the mobile technologies - Standard contextual inquiry - Facilitate enactment by encouraging the user to offer feedback - Alternate between observing and enactment - Co-visioning with the user	
The feature-oriented stigmergy-based collaborative requirements modeling approach aims to provide a feasible solution to the requirements elicitation and evolution problem of Internetware applications, that is, how to elicit the requirements of a large-scale user community with intensive diversity while in the same time continuously evolve the elicited requirements to reflect the latest requirements state of this user community. The main characteristic of this approach is that it does not depend on the requirements engineer to enforce the eliciting and evolving process; on the contrary, it depends on individuals in the user community to elicit and evolve their requirements themselves, through environment-mediated indirect interactions. That is, in the eliciting and evolving process, individuals do not have to communicate directly with each other; they just need to concentrate on their own work, for example, creating new requirement elements in the environment, exploring the environment to view requirement elements having been created, referencing a requirement element if they think it is valuable, and unreferencing a referenced element if they think it is not valuable anymore.	[51]
The Strip Resolution Process (SRP) is made up of the following five elements: (a) A schema is a conceptual abstraction that serves as the basis for ‘‘human information processing, e.g., perception and comprehension, categorization and planning, recognition and recall, and problem-solving and decision-making’’. A schema is the combination of the interpreter’s frame of reference, goals, and plans at a given time. In simpler terms, a schema is the interpreter’s world view. (b) A strip is data that is produced from the interpreter’s observations, discussions with subjects, interviews of the subjects, participation in activities with the subjects, or study of documents. A strip is any phenomenon against which the interpreter tests his understanding (which is based on the interpreter’s existing schema). (c) A breakdown occurs when the interpreter applies his current schema to a strip and finds that he cannot understand the strip. (d) A resolution of a breakdown is the process by which the existing broken down schema is changed to develop a new schema that results in the interpreter understanding the strip. (e) A schema is coherent if it can be applied to many related strips without a resulting break down in the interpreter’s understanding of the phenomena represented by the strips. The following step-by-step process can be considered an instantiation of the SRP for Systems Development: (a) Develop a schema based on observations and discussions with	[45]

customers. (b) Use the schema to make an inference by designing and developing a system functionality for a user requirement as understood using this schema. (c) Test the inference by demonstrating the system to the users, observing users use the system, and discussing with them as to whether the system meets their requirements and work practices. (d) If the test fails, there is a breakdown in understanding and the schema needs to be revised. The revised schema is created by (i) systems developers reflecting on why the break down occurred, (ii) determining what needs to be changed in the current schema to prevent the break down, and (iii) conducting new interviews and observations (if, after reflection and discussion, the reasons for the schema break down are unknown or unclear). (e) If the test does not fail, the schema needs to be tested on other related strips. If all related tests do not fail, then the schema can be considered coherent. The system functionality designed and developed based on this coherent schema can be released to the users.	
Customer Involvement Factory model distinguishes three types of involvement which are: 1. Design for customers: gets input from specific studies of customers such as interviews and incident reports 2. Design with customers: allows customers to react to different proposed design solution such as by letting them rank the importance of requirements 3. Design by customers: get input by having customers providing their own suggestions or describing how they would design a certain part of a system This is done during Customer Participation Session which is a meeting between the customers and the company where ideas are exchanged. Offering feedback after Customer Participation Session ensures the customers their opinions are valued. The Customer Participation Session should not be focused on marketing, but on letting customers share their ideas	[24]
SEEM (Software Engineering Effectiveness Model) consists of the following 5 steps: 1. Justify business case 2. Crease use case description of the business workflow (do not technical words at all) 3. Crease a UML corresponding to the workflow 4. Design the software by mapping the output of 2 and 3 to design model using Model-View-Code paradigm. Afterwards, create a GUI mockup and present to customers. 5. Implementation	[34]