International Journal of Information Systems and Project Management

Volume 6 | Number 4 Article 6

2018

What causes positive customer satisfaction in an ineffectual software development project? A mechanism from a process tracing case study

Bjørnar Tessem
University of Bergen

International Journal of
Information Systems and Project Management
ISSN (print):2182-7796, ISSN (online):2182-7788, ISSN (cd-rom):2182-780X
Available online at www.sciencesphere.org/ijispm

What causes positive customer satisfaction in an ineffectual software development project? A mechanism from a process tracing case study

Bjørnar Tessem
Department of Information Science and media Studies, University of Bergen
Postbox 7802, 5020 Bergen
Norway
www.shortbio.org/bjornar.tessem@uib.no

Abstract:
The customer role is crucial in agile information systems development (ISD). There is, however, a scarceness in research on how this role is enacted, and how its practice influences project outcome. In this longitudinal case study, an agile ISD project is followed with a particular focus on the customer organization's participation, aiming to contribute to the understanding of how customers influence agile ISD projects. The data analysis follows a process tracing approach, a case study method where one aims to identify the causes and outcomes of any kind of process through the rigorous analysis of qualitative data. The analysis of the case shows that the low completion of the initial project requirements was caused by over-scoping and by an immature customer. Further, the customer's acceptance of the outcome was caused by the agile practices introduced in the project. These helped to create a high customer's sense of responsibility for the outcome, which worked as a mediator towards a positive acceptance of the delivery. The study contributes a mechanism for why agile projects may still be successful in light of low delivery. It is also a first case study in the information systems field explicitly using a process tracing approach.

Keywords:
information systems development; agile methods; customer role; process tracing; causal model.

DOI: 10.12821/ijispm060405

Manuscript received: 3 July 2018
Manuscript accepted: 16 October 2018

1. Introduction

Agile software development approaches are common, if not dominating, in information systems project management practice these days. Among the stakeholders in agile projects, the customer has a prominent role as the provider of requirements and domain knowledge to the development team. In the original agile approaches, for instance Extreme Programming [1] or Scrum [2]. The customer is often envisioned as a single person representing the client's interests, being available for the developers to specify and prioritize requirements as well as being an on-site domain expert. In recent practice, particularly as we aim to scale agile methods to many geographically distributed and specialized teams within one project, the customer is often not a single person, but a team or organization of people with the responsibility of providing requirements and quality control [3][4].

The formulation of the Agile Manifesto [5] is mostly a developer-oriented perspective on how to do software development projects, so the developers' practices are in focus, and the customer role is under-specified in the concrete methods. Also in research, the role has not been focused much, but we do know that it is a difficult role to fill, with practitioners enjoying it and exploring a variety of approaches [3]. At the same time it has been observed that customers struggle with the role, leading to project risks [6][7].

Agile software development has become mainstream, and given that customer collaboration is crucial and understudied, it is necessary to establish more knowledge about what effects customer collaboration performance have on the outcome. What are the mechanisms involving customers that enable or inhibit progress and success in agile software development projects? It is the purpose of this article to search for some answers to this question, and to study this, we have analysed qualitative data from a two-year development project, where the customer's performance has had significant effect on the outcome. To get valid conclusions about causes and effects, we have chosen to follow a process tracing case study methodology [8], i.e., a research method where one in a rigorous manner analyses qualitative data to identify mechanisms at work, observed through chains of events, and through this explain the outcome of a case from its particular contingencies.

The outcome of the analysis shows why the outcome of the software development project, in the context of over-scoping and immature customer, still has acceptance from the customer. The answer is that the agile software development methodology, with its communication practices, gradual requirements specification, and continuous delivery, combined with important events in the project increases the customer's sense of responsibility for the outcome, and further causes an acceptance of a low completion of initial requirements. An additional main contribution of the article is that it constitute a first example of how to use process tracing as a research method in information systems and project management research.

The paper is organized as follows: We start with a section on the customer role in agile methods (section 2), motivating the research question, and then continue with a presentation of the particular case study method called process tracing (section 3). After a short presentation of the data collected in this longitudinal study (section 4), we go on with the analysis (section 5), providing examples of how we have used process tracing to explain the mechanisms at work. In a discussion (section 6), we summarize the findings and implications for theory on customer collaboration before we conclude (section 7).

2. The customer role in agile software development projects

An early summary of customer collaboration practice in general is found in a grounded theory study by Martin et al. [3]. They present a number of de-facto roles occurring in customer teams, and in addition, a collection of established customer practices providing value to the development project. In another grounded theory study, Hoda et al. [6] document how projects struggle with lack of customer involvement, identifying causes for this as well as consequences. They also identify a number of "undercover strategies" practiced for increasing customers' involvement, contingent on the project factors.

Some authors have a focus on other aspects of agile methods than the customer role per se, but has findings related to the customer, for instance Ramesh et al. [7] who show that lack of customer competence and lack of customer concurrence imply risks in agile requirements engineering. This is an observation also made in a study by Conboy et al. [9]. A couple of studies emphasize how missing domain knowledge on the developer side causes misunderstanding of requirements provided by the customer [10][11]. Lately there has been attempts to facilitate customer participation in requirements work and decision making suggesting new approaches, techniques, and tools [12]-[15].

There is, so far, not any conclusive studies that the resource demanding customer practices suggested in the Agile Manifesto actually deserve the merit they are given. However, one source of evidence for the necessity of well-functioning customer collaboration is all the research conducted on user participation in software engineering. The user role is not the same as the customer role, but users have the domain understanding and can also assess usability and quality of the resulting products of a development project. Two meta-studies conclude on solid grounds that user/customer participation and/or involvement have positive impact on system success [16][17]. Notice that both these studies include work on customer collaboration in software engineering in their collection of studies. One of the studies includes a collection of user or customer participation practices found in software engineering projects, documenting a large variation on how user participation is managed [16]. Abelein and Paech's [16] meta-study identifies main factors that influence the value of user participation, and human aspects related to user involvement (psychological ownership) and trust are considered important. Siddique and Hussein [18] looks at success criteria for agile projects as seen from the supplier' side, and brings forth the continuous and collaborative supplier-customer assessment of project status as instrumental to success. Worthwhile mentioning is also Nuotilla et al. identification of stakeholder participation and involvement as a main challenge in public domain projects [19]. Another trend supporting the importance of customer collaboration in software development, is in the general project management literature where we now find research that identifies relational contracts combined with good owner-contractor collaboration as factors that contribute to project success [20]-[22].

In a longitudinal case study Bano et al. [23], continuing the previously mentioned meta-study [17], they aim to understand more deeply how user satisfaction, indicating system success, evolves as the contingencies of the project are changing [23]. This kind of deeper analysis of how the theoretical mechanisms contributing to project outcome play out in a particular project, and further how the mechanisms interact, is not found in the research on customer collaboration. Such studies will potentially help to strengthen and adjust existing theories as well as bringing in new perspectives on what makes customer collaboration successful. In the case study presented here, these mechanisms are the main study object, as we identify mechanisms in place in a particular project organization, and analyse how these actually contribute to the outcome of the project.

3. Process tracing case studies

The use of case study research is well accepted in both information systems and software engineering research [24]-[26]. It is a research approach that allows us to get a deeper understanding of theoretical concepts, and in a domain where it is difficult to do valid quantitative studies, the case study is helpful also for theory development. In the domain of customer collaboration in agile software development several of the mentioned studies in the previous section are case studies ([3][6][10][11]).

McLeod et al. [27] focus on longitudinal case studies in software development. They discuss relevant issues for longitudinal research in the domain, for example, access to the case organization during the whole data collection period and collection of data from many sources, representing multiple perspectives. Dubé and Paré [28] have assessed the level of rigor in a large collection of information systems case studies, and found varying rigorous strength. As a consequence, they give recommendations as to how higher rigor could be ensured. In particular, they mention better documentation of data collection and analysis processes.

In political science, there has lately been attempts to formalize case study approaches that aim to understand causes of events and processes. Beach and Pedersen [29] provide an overview of such methods dividing them into three categories, comparative methods, congruence methods, and *process tracing*. The last one, process tracing, is

particularly relevant in cases where you have longitudinal data, and aim to understand what mechanisms work throughout the lifespan of the case, and how they influence the outcome of the process going on inside the case. As such, it matches well with longitudinal studies of software development projects, where one aims to find out what happened, why it happened, and what the consequences were.

Beach and Pedersen [8] describe in detail their perspective on process tracing, and their approach has also been followed in this case study. They see process tracing as a method where a researcher can make strong within-case inferences about why an outcome came about. What were the initial states of a process, what were the events during a process, and how did causal mechanisms at work in the particular domain, contribute to create the outcome? As every case will have its own features and mechanisms in place, the method is a single-case study method. It is essential to note that we are talking about causality in the form on "X caused Y"-propositions, not correlations between variables as often seen in quantitative studies.

Process tracing opens for both deductive and inductive studies. The case may contribute to science by using existing theoretically knowledge, and try to give deeper explanations on how this theory works. But there is also an option to establish new theoretical concepts and statements, verified by good data and analysis, presenting novel causal mechanisms. Often, a process tracing study will have elements of both [30].

Central in process tracing is the concept of a mechanism, which is understood as "an agent or entity that has the capacity to alter its environment because it possesses an invariant property that, in specific contexts, transmits either a physical force or information that influences the behaviour of other agents or entities" ([31]; in [8]). Beach and Pedersen [8] maintain that in the single-case context a mechanism is deterministic, i.e., the effects it produces occur out of necessity in the specific context of the case. Finally, it must be noted that a mechanism is to be seen as something that works on the whole case, and may have some non-trivial complexity relating to variables and causes. It may, however, consist of parts or sub-mechanisms that by metaphor can be seen as "toothed wheels" and "bolts" help to bring about the total effect of the overall "machine" or mechanism.

The analysis in process tracing is based on qualitative data. This can be whole documents, interviews, or even only single statements in an interview or from a meeting's minutes. These chunks of data will through the analysis gain status as evidence. For this to happen, the data must be related to the context and the hypothesized mechanisms, and seen to be relevant for the argumentation. Collier et al. [32] have named evidence in this understanding by the term causal process observations (CPOs).

Identifying the right hypotheses about mechanisms is a challenge, and is depending on theoretical and practical knowledge of the domain under investigation. For example, there could be traces of evidence that relate to a particular explanation from theory, and this would be a starting point for making hypotheses or identify parts that may be included in a hypothesized mechanism. (Sub-)mechanisms explicitly stated or strongly suggested in the data are also candidates.

Arguments in process tracing is rooted in Bayesian logic, where one assign a priori probabilities to statements given no evidence, and then estimate posteriori probabilities based on updating from the use of conditional probabilities relating hypothesis and evidence [8]. This is done in a qualitative manner in process tracing, where conclusions are based on qualitative assessments of the strength of evidence and its power to confirm or disconfirm hypotheses. Van Evera [33] suggests that one categorizes the Bayesian arguments for or against a particular hypothesis into four types, straw-of-the-wind test, smoking-gun test, hoop tests, and double-decisive tests. A straw-of-the-wind test is an application of evidence to strengthen or weaken a hypothesis, but neither at the level of confirmation nor disconfirmation. Smoking-gun tests allow us to strongly confirm a hypothesis, whereas the opposite evidence would not disconfirm the hypothesis. Hoop tests allow us to disconfirm a hypothesis if needed evidence is missing, but not strongly confirm it if the evidence is there. Finally, double-decisive tests are when evidence allows us to confirm one particular hypothesis and disconfirm the rest. Although these categories can be helpful one should be aware that they really can be seen as a division of a continuum with two dimensions, one dimension being the confirmatory power of evidence (because evidence is unique to a hypothesis), and the other dimension being the disconfirmatory power of evidence (because certain evidence is needed to confirm a hypothesis).

Beach et al. [8] emphasize rigor in process tracing, and by that meaning, among others, sources of hypotheses, documentation of evidence sources, prior assessments hypothesis strength, specifications of what kind of evidence one needs for a test to be passed, and documentation and argumentation for the strength of applied evidence. It is challenging to describe all of this at a sufficient level within the space limits of a journal article. Qualitative case studies are by nature demanding to present. However, we will here try to give a convincing exposition of the studied case and the conclusions drawn.

4. The case story and the data

The customer organization of this study has existed for more than 100 years. The last 20 years the company has changed from being mainly involved with services within one particular business domain to be engaged in two related business domains. The old domain (domain O) is rather static, and there are few changes in the domain, whereas the new domain (domain N) is still growing and new approaches and solutions to the business are still developing. The company has a central office in a main, Norwegian city C1, as well as regional offices all around the country. To avoid identification of the organization, respondents and other stakeholders, the description is here kept at an abstract level, but maintaining essential features of the project itself.

The service provided is complex and needs planning before delivery. The company some years ago identified a need for better computer support for their field personnel. The field workers wanted faster access to information, a standardized way of behaving towards customers, and automated documentation of the services provided. Other aims were to present themselves to their clients with a uniform behaviour and being able to collect better statistics about the clients and the market. They conducted an internal process for specifying needs, and an external consultant (A1) was engaged in process modelling for activities to be supported by the final tool. The resulting document was an informal requirements analysis.

During spring 2015, tenders were invited for a 20 months long, 3.0 million Euro software development project. The company has a significant, internal Information and Communication Technology (ICT) department, but they do not have the capacity to run such ISD projects. A small/medium-sized software development company won the bidding process. The contract was formulated as a hybrid contract with three production deliveries, but applying a Scrum process with monthly, intermediate test releases throughout the project. The contractor has its main site in another Norwegian city C2, and a second site in a third city (C3). Both sites were involved in the project. Consequently, we got a distributed project with development teams at two sites, and the company at a third geographical site.

The project started in September 2015, focusing on functionality for domain N. In the first sprints, focus was on going from the specified needs in the informal requirements document with its process models to describe epics and user stories for the domain N in parallel with initial software development. A1 had a significant role in this work together with functional designers at site C3. The customer role was realised by a product owner group that consisted of representatives for both domains, as well as a constituted product owner who lead the group.

During the first six months it became clear that the management for domain N was not satisfied with the solutions for service delivery planning, and the manager took over for the deputy that had participated in the product owner group. The project's contract, due to lacking progress and continuous discussion about changed requirements, was changed into an agile contract in the start of 2016. Simultaneously, A1 was released from the project, and consultant A2 was hired by the company. A2 had experience with agile project practices, and took an internal project manager role. The project also started the development of epics and user stories for domain O during spring 2016. A2 initiated changes to the delivery practice into a continuous, Kanban-like delivery process. Production releases should be delivered every month, as well as bi-weekly releases for testing.

Ten months into the project, they had spent most of the funding for the domain N, but solutions for service delivery planning (main functionality of the system) were not accepted. The intention was now to focus on domain O, but resources were also in the continuation spent on completing domain N. The project was extended with a few months.

During fall 2016, a client for the field workers in domain N was tested in production with positive responses on the users' behalf, even though domain N's main office was dissatisfied because it did not match their ideas of how the business processes should be. Initial functionality was delivered for domain O, as well as documentation of epics and user stories for prioritized functions in that domain. Expectations on the company's side were reduced, but they still trusted that there would be a significant amount of useful functionality at the project's end date. In November 2016, a report stating that with current development speed less than half of the first priority features of the system could be delivered within the project time. The company representatives in general did not seem to a serious extent to be critical to the project process and the contractor, and still had fairly high expectations.

The project was prolong until the end of 2017, when the project went into maintenance status. During that period, not many significant changes happened to the organization of the project. One person in domain N was from late 2016 partially assigned responsibilities to participate in developing services related to the customer's own clients' planning and accounting of production. Functionality for this was delivered in 2017. Functionality regarding plans for customer interaction in domain O was also delivered during 2017. The customer's way of relating to the contractor did not change much, even though variants of how to handle requirements were tried and shifted out. They maintained a good cooperation towards the end of the project.

The data from this project are from the whole project period, and are mainly of four types, interviews, meeting minutes, status reports, and meeting observations. The bulk of the data are from the two first categories, and in particular the interviews have more explanatory power.

The 27 semi-structured interviews were done starting in March 2016 and continuing with some longer breaks up until October 2017. The interviews lasted from 40 minutes up to an hour. Ten of the interviews were done with three members of the developer teams, two of them in city C2, a software architect and a test manager, and one in city C3, a functional architect (designer). The rest of the interviews (17) were with the company's project representatives, located in city C1. The roles of the these interviewees were a product owner, the internal project manager, two business domain middle managers, the project owner (the Chief Information Officer (CIO), the leader of the local ICT department), two ICT department developers, and two domain specialists. The topic of the interviews was all the time on project progress and the reasons for events and outcomes of the project. In the last eight interviews we also talked about the effect of agile methods and customer-contractor communication in the project. All the interviews were conducted by this article's author. We also have data in the form of documents with minutes from project manager meetings (a total of 80 meeting minutes), steering groups (22), work group meetings (15), and retrospectives (25). In addition we have had access to project status reports produced from June 2015 up to and including September 2017 (21 reports). Finally, we have six observation protocols from a graduate student, who was taking notes at retrospective meetings in the start of the project up until March 2016. The student then moved on to collect data from a different case, irrelevant for this study.

5. Analysis

What makes this project particularly interesting was the fact that the customer more than a year into the project was informed that they would get less than 50% of their initial first priority features, and at the same time was not very critical to the contractor's work. Instead of a situation where the parties would play a blame game, the customer and the contractor seem to have been in agreement to make the best out of the situation. So we have a conjunctive outcome o that we want to explain here, formulated as $o = o_1 \wedge o_2$ where o_1 = "< 50% delivery" and o_2 = "accepting customer". What is it with this particular project that has caused this seemingly contradictory situation? Can we find the mechanism behind o by identifying events and their consequences related to the customer's participation in the project?

5.1 Generating mechanism hypotheses

Establishing which hypotheses one wants to test in process tracing is one of the challenges of the method. Guidelines say that we could look for potential hypotheses in theory, and try to match these with observations found in the data. For example, there may be statements in interviews that actually try to explain causalities. Other empirical reports on

similar topics, and the researcher's own experience and knowledge of the domain may also provide starting points for hypothesis generation. Thus, we can identify potential candidates for mechanism parts, which may in the end be combined to provide a complete explanation. In this work, candidate hypotheses originate from statements in the interviews as well as other empirical studies and theoretical knowledge about relations between variables and system success.

A rigorous start would be to first use data to verify the outcomes o_1 and o_2. There is a strong smoking-gun evidence that o_1 is true based on the report that actually states this. Later support of this is that at towards the project's end we find that only requirements regarding information presentation and data consistency control has been delivered to domain N, missing support for field work. In domain O information presentation and only parts of the complex field work support are delivered. This is far below the initial specification. As for o_2 = "accepting customer", this would have a comparably small prior probability ($P(o_2|o_1)$) in this case. But there is smoking-gun evidence found as the customer in the contractual change actually took most of economic responsibility for the slow project progress. Other supporting evidence is this sentence about the CIO found in a project status report about midways in the project: *"It was especially nice that the CIO [the customer's CIO] dropped in to say hello to the [the contractor's] team, and said that they [the customer] were satisfied with the work done."* A successful hoop test to support an alternative to o_2 would need evidence that showed that the customer at some time has tried to place economic and legal responsibility on the contractor. This is not found, so o_2 is considered to be verified. After this verification of the main outcomes, it would be interesting to establish the mechanism that results in $o = o_1 \wedge o_2$ even in a situation where the probability of o_2 given o_1 ($P(o_2|o_1)$) is small.

An standard process tracing approach for developing the mechanism that explains the case is to try to verify parts that would have to go into that mechanism, and then try to construct a complex mechanism where these parts work together. The parts we can hypothesize being at work in this case are related to those variables shown to contribute to success in systems development projects [34]. A requirement is also that we in fact have qualitative data enabling us to analyse these variables. The focus of this research was on the customer role, so we have chosen the customer's capabilities and behaviour, the contractor's capabilities and behaviour, the technological complexity of the system, the project management practices including user participation and communication practices, and finally project scoping as important variables to study. We will in the analysis describe the sub-mechanisms that has been in play in this case, and finally construct a complete mechanism from those partial mechanisms.

5.2 Sub-mechanism 1: Over-scoping and customer capabilities cause low delivery

In this section, there is an analysis of what actually lead to low delivery, i.e., o_1. First, there is an analysis of possible causes, and then an analysis on how they interact to create a low degree of project completion.

The most visible evidence that scope size has had an impact on outcome comes from the re-estimation report produced towards the end of 2016. This report itself is a CPO on outcome o_1, indicating that the hypothesis h_1="Initial project over-scoping caused o_1" needs to be tested. An argument for over-scoping is that the project consisted of two equally sized parts. At the time of the re-estimation report there had been progress for domain O, and almost half of the project period still was left. Then, if adequately scoped initially, at least 50% of the total scope, i.e., all of domain O, and in addition the accepted functionality for domain N, would be expected to be delivered. This is considered a double decisive evidence that supports h_1 as part of the explanation for over-scoping. We have no clear cut explanations as to why the project was initially over-scoped, but lack of understanding of technological and organizational complexity could be one, in combination with inadequate funding to the project. In an interview, a respondent mentioned high expectations on the possibility to integrate off-the-shelf software in the solution as a possible cause.

The customer or user's ability to provide sufficient support to the development team in a system development project has been critical in many instances, and lack of participation and engagement in the project reduces system success [35]. In this case, there is lots of evidence pointing towards insufficient customer participation. There are two main reasons for this: immaturity or lack of experience in the organization when relating to system development projects, but also low capacity for contributing to the project. This leads to the proposal of hypothesis h_2 = "Low customer capability

caused o_1". A statement from a contractor interviewee early in the project is clearly indicating this: "*…one thing to talk about is process competence - process and maturity with regard to develop IT systems. I would say that they are rather immature with regard to developing IT systems. They don't seem to have good heuristics about how behave as customer in relation to a contractor, for instance on being able to deliver the domain knowledge, to ensure that one has resources available for long enough time, and that your willing to organize a set up for participation that facilitates the process to run together with us."*

To prove this relation it is important to search in data for CPOs and events that gives a causal chain from low customer maturity to o_1. CPO-s that could support this chain, would report on low management support, conflicting priorities, unengaged users, missing understanding of the project goals, etc.

There is additional evidence, besides the statement above, which strongly supports the fact that the customer had low capability to handle this kind of projects, but only for domain N and the customer's ICT department. Domain O defined their goals clearly, for instance, by having described established business processes that they will support. They also have given very high priority to the project, by assigning qualified personnel to work 100% as customer representatives.

When it comes to evidence regarding domain N, they have since the start of the project to the last few months got clear messages about their participation. This is visible in the project management meeting minutes from the beginning of the project and even 5 months before the end we find in a document: *"The project is still not sufficiently prioritised in the domain N department"*. The middle manager of domain N who was responsible for the domain's participation was more critical to the way the project was run overall, but also he admitted that resources was not sufficient: *"So we may also be criticized for not having invested enough in the project, that we did not have enough personnel to ensure quality."* Priorities were not the only problem for domain N, as unsettled business processes lead to a situation where consultant A1 and the contractor's requirements team focused on the field workers' current practices. The initial focus on user needs obviously have support in the literature on information systems development. However, the users' immediate need was obviously in conflict with the intention of the main office's wish to establish new business processes and to support those instead. This lead to continuous disagreements about requirements, and finally domain N did not accept the solution for field support.

The customer's ICT department also had a significance role in this project, as they were responsible for the databases and API-s for accessing the data. They seem to have delivered the needed solutions only partially or after repeatedly being pushed for response. There is a constant focus in meeting minutes on the need for the ICT department to prioritize the provision of APIs to the project. Particularly towards the end of the project, this becomes visible in the data, both in interviews and minutes. This is mainly a priority issue, as the consultant A2 expressed: *"Even though they do understand agile, it has not really been understood that they actually must assign resources to this. They blame it on priorities; that business management requires things that they have to give priority."*

The contrast to department O is pronounced. In interviews, their participation is praised both in terms of resources assigned and their ability to respond constructively. One interviewee said, *"The work that has been done by domain O and those who have participated there has been fantastic, and there we have indeed come a long way".*

The maturity of the contractor is known to be a main factor in system success. This is for instance the perspective taken in the Capability Maturity Model (CMM) [36], and in models that suggest approaches to leverage an agile team's maturity [37]. There were some traces of dissatisfaction with the contractor in the later interviews. For example, the consultant A2 expressed that they are not working in an agile manner. They also continued work with user stories and epics of parts of the system that were supposed to be postponed. The contractor did probably not have the most mature agile team, but there no evidence that lack of contractor maturity caused o_1. On the other hand, the contractor seemed to have processes that to some extent match recommended processes for developing own maturity [37], suggesting awareness of the organization's need to improve.

The last factor looked at is technological complexity. There is not much evidence in the data that the technology was too difficult to handle for the contractor or the ICT department. There was some discussions regarding the data

organization that were solved after some negotiations. In addition a third party service provider was slow at responding to requests for APIs to their services, but these issues also seem to have been solved.

When it comes to the causality, of course over-scoping in itself is an important cause for low delivery. In addition, there are three main observations regarding domain N and the ICT department. First, they have had a lack of resources to participate well in the process. Second, there has been conflicting priorities, and finally, the business processes were not settled for domain N.

Thus, the customer's immaturity contributed to reducing the amount of finished functionality in addition to the effect of over-scoping. Large parts of the solutions that was made in the start of the project was in the end not accepted by domain N due to disagreement on requirements within domain N. This obviously contributed to low delivery. Lack of internal resources in the domain lead to inability to define new business processes in time, and because of conflicting priorities, representatives for domain N did not participate in or was not prepared in planning meetings. Dissatisfaction with lacking prioritisation was repeatedly expressed in minutes from retrospective meetings. As the main office did not contribute enough, the contractor relied on information from domain N's field workers. Requirements thus were often in conflict with the main office's wishes, and postponement of prioritization lead to little work to do for the system developers, making them unproductive in addition to having made the wrong solution. A quote showing the project's lack of agreements and communication within domain N, was when one representative from the contractor said about a manager in domain N: *"He was included because he had strong opinions about things, in requirements work or in development, and then he suddenly said «This is not how it should be». And this happened after some of his subordinates in fact had given their support for this solution."* Another event that shows lacking priorities in action was a superior manager's prioritization of a business report, causing the domain N representative to be unprepared for planning meetings (if at all participating).

The data also verifies that the ICT department's ability to provide the APIs to the databases in time has caused delays in the project. An issue that also was mentioned frequently, was delays in deployment of tested solutions into production. This was evidently caused by low capacity in the ICT department. A quote from several meeting minutes summarize the ICT department's lack of capacity: *«Lacking capacity to handle changes and new challenges in the ICT department is a high risk».*

5.3 Sub-mechanism 2: over-scoping and a customer with lacking capabilities, combined with agile project management practice causes customer sense of responsibility

In a case study focusing on reasons for too large scope, Bjarnason et al. [38] identify several sources of over-scoping in the project, the most prominent cause being the continuously changing requirements. In their case, approaches like agile cross-functional teams, continuous prioritization, and gradual detailing of requirements were applied, and were shown to be helpful strategies for managing over-scoping in the project. The case in this article also has an agile approach to project management, and also what may seem like a positive outcome of the project, making it similar to Bjarnason et al.'s case. Bjarnason et al. refers to the concept of "communication gap" between developers, requirements workers, and the customer, and suggest that the agile practices reduce the communication gap and thus creates acceptance for less than expected delivery.

This may suggest the sub-hypothesis "The agile project management approach facilitates outcome o_2 in the context of over-scoping and customer with lacking capabilities" for further analysis. Still, even though acceptance may be an outcome, there is a lack of understanding of how agile project management may contribute to acceptance. What are the mediating mechanisms? Closing of communication gaps is one of the goals and effects of agile methods, and it was obtained in the case in this article as well. However, it is not sufficient as mediating variable to explain acceptance. Why do closed communication gaps cause satisfied customers? As a sparking clue to how this analysis should proceed one of the interviews said *"if you have a* ***common responsibility*** *to deliver as much as possible, then you will become more interested in the process, and you see what is the best way to go"*. This suggests that the agile practices in our case in fact may have caused higher trust between customer and contractor, leading to a higher acceptance of own responsibility for the outcome of the development process. In light of this, it is suggested that o_3 = "customer sense of

responsibility" could be a mediating variable towards acceptance of the outcome. In the remaining of this subsection, we will try to verify the sub-hypothesis h_3 = "The agile project management approach facilitates outcome o_3 in the context of over-scoping and customer with lacking capabilities".

There are several clues showing that the customer has had a large sense of responsibility. For instance, after domain N's insufficient participation in the initial stages of the project, combined with a growing understanding of the complexities of their original requirements, a change of contract was initiated by the CIO of the customer. This changed the project from a fixed price hybrid project to an agile project with continuous delivery and with less focus on initial requirements. The customer in this manner took most of the cost relating to unsuccessful progress. At the same time, an internal project manager (A2) with long-time agile experience was hired to support the initial product owner who had little experience with IT projects. The agile project manager championed continuous delivery even more, and iterations were cut down to two weeks. In addition, requests for dedicated support in the project owner group were sent both to domain O and domain N. Domain O delivered one person 100%, which was a success. Domain N dedicated less in terms of work hours, but appointed a person who supported with domain knowledge related to a specific functionality (control of complex production data). This data control functionality was delivered to domain N towards the project termination, and this was considered a minor success in the project. The customer gradually assigned more and more resources to the project, responding to the contractor's requests for domain knowledge and consistent requirements prioritization.

The causes for the sense of responsibility we have observed in this project could be other than the agile project management practice. For example, it could be that the customer organisation could have a high sense of responsibility for the outcome from the start, causing them to invest more in the project than initially planned when problems occurred. To validate the hypothesis here, that the project management practice actually was causal here, we need data that shows how agile practices have changed the overall sense of responsibility of the customer organisation.

A first observation is that the communication practices of agile methods have been instrumental in this, even though a fixed price contract was initially in the foundation of the project management practice. From the start, frequent meetings were held, always in a cordial atmosphere; most meeting minutes throughout the project report that project communication is good. In the minutes it is also evident that the contractor from very early pointed at the insufficient and inconsistent participation of domain N. Internal representatives in the project confirms this in the early interviews.

As normal in agile development projects, domain knowledge and precise specifications of the requirements were also issues in communication. Early on it became clear that the requirements were not well specified, which again lead to many change requests in the project. Negotiations on change took much time, which had consequences for progress: *"Last year there was a lot of standoffs about change. We had to decide how to handle the changes, which not necessarily need to be handled in an agile manner. But then you must negotiate on each change, so I suggested we left the perspective that change was unwanted, and instead consider it wanted. How can we make this, and not spend a lot of time on deciding if this is a change."*

One communication practice that build trust in the contractor was the monthly demo meetings that showed the progress of the project, allowing the whole organisation to reflect and propose improvements. Many showed up at these meetings, contributing to a well-rooted project in the customer organization. Other straw-in-the-wind evidence suggest that the contractor got a high trust with the customer, leading to respect for their opinions, and further to an understanding of own responsibilities in the project. Communication in the O domain seemed to have been particularly successful. The domain O single contact point said: *"I have very good collaboration with them, and I must say I'm surprised about how fast they grasp my domain and understand how it works, so I'm very impressed. So I think this collaboration works fantastic."* A representative from the contractor in C3 expressed: *"The other thing we did to make things work, is that he [the domain O expert] has been here weekly on fixed days, and we have been in C1 and worked together. We have a very close collaboration, because we have been a lot together. We have done social things together, so it has been very good."*

Agile communication practices did cause the change to an agile contract, and with that, also established more focus on continuous work with requirements and delivery. The agile practice of gradual requirements work seem to have contributed to a better understanding of the customer's own role. One domain O specialist said: *"They continuously*

force us to be precise in our requirements. And we were not really aware of our needs beforehand, so that is positive." Finally, the practice of continuous delivery has helped the customer representatives to understand the value of own participation. In domain N, the successful delivery of production data control was a result of continuous improvements based on user experiences: *"We have adjusted that model quite a lot, because one has got a lot of feedback from those who have tested it."* The gradual inclusion of new functionality, also lead to an understanding of poor data quality in the customer's own databases, initiating processes to improve their data as a means to better the value of the delivered software.

5.4 Sub-mechanism 3: low delivery and customer sense of responsibility causes customer acceptance

To finalize our analysis we need to continue the development of the sense of responsibility variable and see how it causes the customer's acceptance of the outcome, and verify the sub-hypothesis h_4 = "low delivery and customer sense of responsibility causes customer acceptance" or shortly stated "$o_1 \wedge o_3$ causes o_2". Several statements from interviews contribute to the verification of the sub-hypothesis: For example,e project manager/product owner A2 said *"I believe that without the dedicated specialists in domain O, then they would not have been close to the solution they now have on the field client. It is absolutely necessary. I think that has more of an influence than the contract."* A developer said: *"We have instead worked with real priorities and what is important. The trust has been deciding for the outcome."* The middle manager of domain N said: *"We have been realistic, so to the extent that they do this to ensure quality in what we get, then that is the most important thing."* The statements are isolated, perhaps by themselves not enough to prove that the customer's sense of responsibility that mainly creates the acceptance of the outcome. But then again, following a Bayesian logic, would it be probable that other hypotheses could lead to the statements above and many other similar statements? For example, would we have observed these statements regardless of the project management approach and/or with a sense of responsibility not influenced by agile practices? A yes to that question is hardly defendable. The conclusion is that the sub-hypothesis is verified.

The complete mechanism showing the causal links between variables is given in Figure 1. The arcs connecting the causal links represent Boolean and-operators, also indicating how the three sub-mechanisms verified in the analysis work.

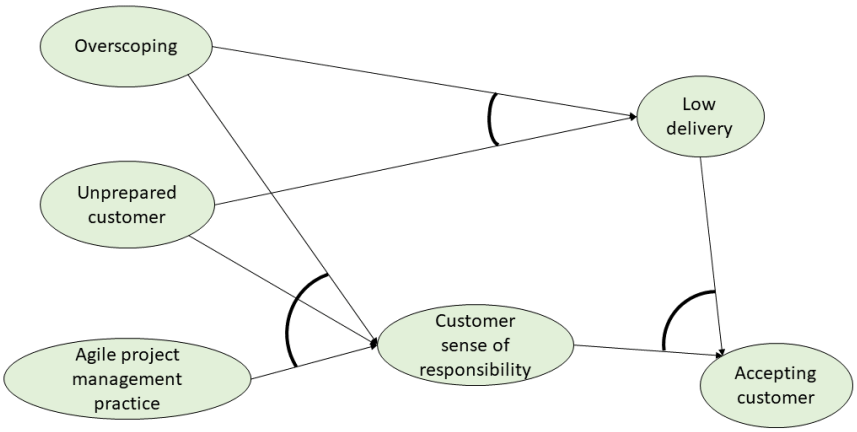

Figure 1 The complete mechanism leading to a software development project with low delivery, but an accepting customer

6. Discussion

The case presented in this study has two interesting outcomes: o_1 = "< 50% delivery" and o_2 = "Accepting customer". Often o_1 would lead to the negation of o_2, with consequential discussion of blame, negotiations of economic responsibility, etc. But here o_1 and o_2 co-occur. As we have documented in the analysis, the causes for o_2 in the light of o_1 is an agile project management approach, leading to a sense of responsibility on the customer side which again leads to acceptance of the final outcome. Figure 1 summarizes the mechanism at work in the project, focusing on important contingencies in the project. Events occurring as an outcome of the described mechanism, and that have been particularly significant to obtain o_3 = "customer sense of responsibility" and further o_2 = "Accepting customer" are:

1. Change of contract from fixed-price to agile (spring 2016);
2. Hiring consultant A2 to work as a product owner (spring 2016);
3. Change of project management approach towards more continuous delivery (spring 2016);
4. Repeated requests from the contractor about more contributions from customer (whole project period);
5. Customer assigning local domain experts with clear responsibilities to the project (early fall 2016).

The case study here verifies observations on the value of developer-customer communication in ISD [39], but also indicate a more precise cause for why communication can be important, namely making the customer aware of the competencies expected from themselves to make the project successful. It is also an example that partially matches Bjarnason et al.'s finding that agile methods contribute to a shared acceptance of the lack of delivery due to over-scoping [38].

The concept "customer sense of responsibility" thus represent a theoretical contribution in the understanding of how the customer may respond to agile information system practices. The sense of responsibility is a mediating variable that brings about a higher chance for acceptance of the final result and also seems to result in more engagement and contribution in the project during the development project.

The process tracing methodology applied here is an attempt to use qualitative data to establish truths about single cases. Like quantitative methods, it is not infallible, but in light of existing theories and previous empirical findings, other researchers' interpretation of the qualitative data most likely would support an explanation model similar to the one presented in Figure 1. Some links are not as strong as other, parallel to significance levels in numerical studies. But in this case the causal links must be considered to be strongly verified. Process tracing has not been tried out in information systems research before, and in this article we have documented how it can be used and further how it can contribute to an increased understanding of the information systems development field. It is particularly interesting that it, as in this case, has the potential to both verify existing theory as well as contributing with new theoretical concepts and relations.

In a case study on a medium sized ISD project, where you analyse system success, data is personal and sensitive for the respondents. Project failure and success may be attributed to persons who have no wish to be public persons. The researcher may need to anonymize not only persons, but also the project, and even the domain of the project. This is a challenge also observed by Milne and Maiden in analysis of power relations in requirements engineering [40]. In this article, even the business domain is anonymised, perhaps reducing the validity and usefulness of the study. Privacy has been prioritized, and the analysis has been presented at this abstraction level to ensure that the anonymity of respondents.

For the practitioner side, the presented story further adds to the evidence that all stakeholders, and in particular the customer itself, need to be aware the customer's particular responsibility for a project to be successful. Problems encountered in the cooperation need to be handled with flexibility and competence. The case illustrates that if there are obvious and early problems with the requirements in a fixed price ISD project, quickly changing to an agile approach may be an approach to reduce conflict levels and support a delivery that still can be considered successful.

7. Conclusion

In this case study we have seen how particular features of an ISD project have contributed to the outcome of the project. The process tracing method as applied gives us an insight into how one could use established theoretical knowledge and established empirical findings together with own empirical data to understand how the events and mechanisms in an ISD project actually influence the outcome. The concrete result is a mechanism for variables and outcomes valid in the particular case. Within this model, a significant contribution is the importance of the concept "Customer Sense of responsibility" as a mediator of the effects of agile project management practices. For project management practice, this study suggests an approach to explaining project outcomes in light of initial contingencies, project events, and their interactions. In particular, we have observed that participation in an ISD is demanding for a customer, and the customer's maturity level and other competencies should be in focus to ensure project success. This case study confirms that weak customer competence can be alleviated by abiding to agile practices, as these practices may bring about a mutual understanding of the project among the stakeholders.

Qualitative case studies are based on the researchers' interpretation of data as well as being problematic to generalize. So any theoretical ideas would be preliminary. In this case, it is the idea that "customer sense of responsibility" is a human aspect, alike to user involvement and trust, that contributes positively to system acceptance. To further validate the findings of the case, one needs additional case studies with agile customers. Such studies could aim to investigate the generalizability of the causal relations and the total mechanism found in this case, including the usefulness of the "customer sense of responsibility" as a theoretical concept. In the long term, quantitative studies aiming to verify this and other general theoretical mediators of customer effects in agile ISD should be conducted.

Acknowledgments

The presented research was partially supported by the Agile 2.0 project funded by the Norwegian Research Council (Project no. 236759).

References

[1] K. Beck, *Extreme Programming Explained: Embrace Change,* 1st ed. Reading, MA: Addison-Wesley, 2000.

[2] K. Schwaber and M. Beedle, *Agile Software Development with Scrum*, 1st ed. Upper Saddle River, NJ: Prentice Hall PTR, 2001.

[3] A. Martin, R. Biddle and J. Noble, "An ideal customer. A grounded theory of requirements elicitation, communication and acceptance on agile projects," In *Agile Software Development: Current Research and Future Directions*, T. Dingsøyr, T. Dybå, and N.B. Moe, Eds. Berlin: Springer-Verlag, 2010, ch. 6, pp. 111–141.

[4] J. M. Bass, "How product owner teams scale agile methods to large distributed enterprises," *Empirical Softw. Eng.* vol. 20, no. 6, pp. 1525–1557, December, 2015.

[5] *Manifesto for Agile Software Development* [Online (2018, July 3)]. Available: http://agilemanifesto.org

[6] 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.

[7] B. Ramesh B, L. Cao L and R. Baskerville, "Agile requirements engineering practices and challenges: an empirical study," *Info Systems J*, vol. 20, pp. 449–480, 2010.

[8] D. Beach and R.B. Pedersen, *Process-tracing methods: Foundations and guidelines.* Ann Arbor, MI: University of Michigan Press, 2013.

[9] K. Conboy, "Agility from first principles: reconstructing the concept of agility in information systems development," *Info. Sys. Res.*, vol. 20, pp. 329–354, 2009.

[10] M. Pikkarainen, J. Haikara, O. Salo, P. Abrahamsson and J. Still, "The impact of agile practices on communication in software development," *Empirical Softw. Eng.*, vol. 13, pp. 303–337, 2008.

[11] M. Korkala, P. Abrahamsson and P. Kyllonen, "A case study on the impact of customer communication on defects in agile software development," In *AGILE '06: Proceedings of the Conference on AGILE 2006*, Washington DC: IEEE Computer Society, 2006, pp. 76–88.

[12] I. Otaduy and O. Diaz, "User acceptance testing for Agile-developed web-based applications: Empowering customers through wikis and mind maps," *Journal of Systems and Software,* vol. 133, pp. 212–229, 2017.

[13] E. Bjarnason, M. Unterkalmsteiner, M. Borg and E. Engström, "A multi-case study of agile requirements engineering and the use of test cases as requirements," *Information & Software Technology,* vol. 77, pp. 61–79, 2016.

[14] E. Bjarnason and H. Sharp, "The role of distances in requirements communication: a case study," *Requir. Eng.*, vol. 22, no. 1, pp. 1–26, 2017.

[15] S. Dimitrijevic, J. Jovanovic and V. Devedzic, "A comparative study of software tools for user story management," *Information & Software Technology,* vol. 57, pp. 352–368, 2015.

[16] U. Abelein and B. Paech, "Understanding the influence of user participation and involvement on system success – a systematic mapping study," *Empir Software Eng,* vol. 20, pp. 28–81, 2015.

[17] M. Bano and D. Zowghi, "A systematic review on the relationship between user involvement and system success," *Information and Software Technology,* vol. 58, pp. 148–169, 2015.

[18] L. Siddique and B.A. Hussein, "A qualitative study of success criteria in Norwegian agile software projects from suppliers' perspective," *Intl. Journal of Information Systems and Project Management,* vol. 4, no. 2, pp. 63–79, 2016.

[19] J. Nuotilla, K. Aaltonen and J. Kujala, "Challenges of adopting agile methods in a public organization," *Intl. Journal of Information Systems and Project Management,* vol. 4, no. 3, pp. 65–85, 2016.

[20] M. Suprapto, H.L.M. Bakker, H.G. Mooi and W. Moree, "Sorting out the essence of owner–contractor collaboration in capital project delivery," *International Journal of Project Management,* vol. 33, no. 3, pp. 664–683, 2015.

[21] M. Suprapto, H.L.M. Bakker and H.G. Mooi, "Relational factors in owner–contractor collaboration: The mediating role of teamworking," *International Journal of Project Management,* vol. 33, no. 6, pp. 1347–1363, 2015.

[22] M. Suprapto, H.L.M. Bakker, H.G. Mooi and M.J.C.M. Hertogh, "How do contract types and incentives matter to project performance?," *International Journal of Project Management,* vol. 34, no. 6, pp. 1071–1087, 2016.

[23] M. Bano, D. Zowghi and F. da Rimini, "User satisfaction and system success: an empirical exploration of user involvement in software development," *Empirical Software Engineering,* vol. 22, no. 5, pp. 2339–2372, 2017.

[24] I. Benbasat, D.K. Goldstein and M. Mead, "The Case Research Strategy in Studies of Information Systems," *MIS Quarterly,* vol. 11, no. 3, pp. 369–386, 1987.

[25] G. Walsham, "Interpretive case studies in IS research: nature and method", *European Journal of Information Systems,* vol. 4, no. 2, pp. 74–81, 1995.

[26] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering,* vol. 14, no. 2, pp. 131–164, 2009.

[27] L. McLeod, S.G. MacDonell and B.Doolin, "Qualitative research on software development: a longitudinal case study methodology," *Empirical Software Engineering,* vol. 16, no. 4, pp. 430–459, 2011.

[28] L. Dubé and G. Paré, "Rigor in Information Systems Positivist Case Research: Current Practices, Trends, and Recommendations," *MIS Quarterly*, vol. 27, no. 4, pp. 597–636, 2003.

[29] D. Beach and R.B. Pedersen, *Causal Case Study Methods. Foundations and Guidelines for Comparing, Matching, and Tracing.* Ann Arbor, MI: University of Michigan Press, 2016.

[30] C. Trampusch and B. Palier, "Between X and Y: how process tracing contributes to opening the black box of causality," *New Political Economy*, vol. 21, no. 5, pp. 437–454, 2016.

[31] D. Waldner, "Process tracing and causal mechanisms," In *The Oxford handbook of the philosophy of social science* H. Kincaid, Ed. Oxford: Oxford University Press, pp. 65–84, 2012.

[32] D. Collier, H.E. Brady and J. Seawright, "Sources of Leverage in Causal Inference: Toward an Alternative View of Methodology," In *Rethinking Social Inquiry: Diverse Tools, Shared Standards*, 2nd ed., H.E. Brady and D. Collier, Eds., Lanham, MD: Rowman and Littlefield, pp. 161–99, 2010.

[33] S. van Evera, *Guide to Methods for Students of Political Science.* Ithaca, NY: Cornell University Press, 1997.

[34] R. Irvine and H. Hall, "Factors, frameworks and theory: a review of the information systems literature on success factors in project management," *Information Research*, vol. 20, no. 3, 2015. Paper 676. Available: http://InformationR.net/ir/20-3/paper676.html

[35] Y.W. Hung, S.C. Hsu, Z.Y. Su and H.H. Huang, "Countering user risk in information system development projects," *Intl Journal of Information Management*, vol. 34, pp. 533–545, 2014.

[36] E.G. McGuire, "Factors affecting the quality of software project management: An empirical study based on the Capability Maturity Model," *Software Quality Journal*, vol. 5, pp 305–317, 1997.

[37] R.M. Fontana, V. Meyer, S. Reinehr and A. Malucelli, "Progressive Outcomes: A framework for maturing in agile software development," *Journal of Systems and Software*, vol. 102, pp. 88–108, 2015.

[38] E. Bjarnason, K. Wnuk and B. Regnell, "Are you biting off more than you can chew? A case study on causes and effects of overscoping in large-scale software engineering," *Information and Software Technology*, vol. 54, pp. 1107–1124, 2012.

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

[40] A. Milne and N. Maiden, "Power and politics in requirements engineering: embracing the dark side?," *Requirements Engineering*, vol. 17, pp. 83–98, 2012.

Biographical notes

Bjørnar Tessem
Bjørnar Tessem is Professor in information science at the Department of Information Science and Media Studies, University of Bergen. His research and teaching cover a large span within information science, including empirical studies of information systems development and agile software development. He has also done both theoretical and applied research in artificial intelligence and machine learning, as well as the design of new media with an emphasis on location-based media and interaction design. He has published in high-ranked journals and conferences in the whole breadth of his research. Professor Tessem also has an adjunct professor position at the Department of Strategy and Management, Norwegian School of Economics where he teaches artificial intelligence and machine learning.

www.shortbio.org/bjornar.tessem@uib.no