Published in Proceedings of the 19th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2013), Essen, Germany

@INPROCEEDINGS{conf/refsq/RempelMKP13, author = {Patrick Rempel and M{\"a}der, Patrick and Tobias Kuschke and Ilka Philippow}, title = {Requirements Traceability across Organizational Boundaries-A Survey and Taxonomy}, booktitle = {Proc. of the 19th Int. Working Conf. on Requirements Engineering: Foundation for Software Quality (REFSQ 2013), Essen, Germany}, year = {2013}}

Requirements Traceability across Organizational Boundaries - A Survey and Taxonomy

Patrick Rempel, Patrick Mäder, Tobias Kuschke, and Ilka Philippow

Department of Software Systems, Ilmenau Technical University
{patrick.rempel|patrick.maeder|tobias.kuschke|ilka.philippow}@tu-ilmenau.de

Abstract. [**Context and motivation**] Outsourcing of software development is an attractive business model. Companies expect cost reduction, enhanced efficiency, and exploited external resources. However, this paradigmatic shift also introduces challenges as stakeholders are spread across distinct organizations. [**Question/problem**] Requirements traceability supports stakeholders in satisfying information needs about developments and could be a viable way of addressing the challenges of inter-organizational development. While requirements traceability has been the subject of significant research efforts, its application across organizational boundaries is a largely unexplored area. [**Principal ideas/results**] We followed a qualitative research approach. First, we developed a taxonomy identifying the needs of inter-organizational traceability. Second, we conducted semi-structured interviews with informants from 17 companies. Eventually, we applied qualitative content analysis to extract findings that supported and evolved our taxonomy. [**Contribution**] Practitioners planning and managing inter-organizational relationships can use our findings as a conceptual baseline to effectively leverage traceability in those settings. Effective traceability supports projects in accomplishing their primary goal of maximizing business value.

1 Introduction

Requirements traceability has been commonly recognized by researchers and practitioners alike as critical element of a rigorous software development process[1]. Gotel and Finkelstein defined requirements traceability as the ability to describe and follow the life of requirements, in both a forwards and backwards direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through all periods of on-going refinement and iteration in any of these phases) [1]. Software process initiatives, such as CMMI, formulated the goal of maintaining bidirectional traceability of requirements. Requirements traceability supports a stakeholder in satisfying information needs within a software development process. The applicability of this concept has been studied for ordinary software development projects. However, as software projects become bigger and bigger there is a tendency to outsource parts of the software development process [2].

[1] http://www.coest.org/index.php/what-is-traceability (accessed: October, 2012)

In IT outsourcing the software development process is distributed across two types of actors – clients and suppliers. In a simplified view, a client produces a requirements specification. A supplier implements the software product according to the client's requirements specification. This form of IT outsourcing offers advantages like leveraging external IT assets [3] and capitalizing the global resource pool [2]. Due to these advantages, outsourcing is a commonly applied IT strategy pattern. However, beside these advantages the organizational border between two or more cooperating actors produces a distance. This distance leads to complexity risks in the software development process that need to be bridged [4]. Traceability could be viable way of bridging inter-organizational distance.

Only little empirical knowledge is available on the impact of IT outsourcing on requirements traceability. We followed a qualitative research approach to close that gap. First, we developed a taxonomy to identify actors and their interactions in inter-organizational projects. Second, we conducted semi-structured interviews with informants from 17 companies to understand the impact of inter-organizational software projects on requirements traceability. Eventually, we applied qualitative content analysis to extract findings that supported and evolved our taxonomy. We discussed needs for requirements traceability from the perspectives of actors in inter-organizational projects.

Our paper is organized as follows. Section 2 reviews related work in the area of requirements traceability and inter-organizational software projects. In Section 3, we propose a taxonomy of actors and interactions in inter-organizational outsourcing projects based on a literature study. This conceptual framework was used to plan and conduct the interview study, which is described in Section 4. In Section 5 we discuss the results of our study, which we extracted via qualitative content analysis from the captured interview minutes and field notes. Section 6 discusses possible threats to the validity of our work and how we mitigated them. Finally, Section 7 concludes our work and outlines future research directions.

2 Related Work

Several studies examined the general application of requirements traceability in software development projects. Gotel and Finkelstein [1] studied requirements traceability practices and highlighted especially the demand for supporting pre-requirements traceability. Ramesh and Jarke [5] conducted intensive interview studies with practitioners. As a conclusion they proposed two traceability reference models. Arkley and Riddle [6] conducted a survey and explored the so-called traceability benefit problem. This problem arises as trace recorder and user are typically different stakeholders. We also conducted a survey on how requirements traceability activities are embedded into company processes [7]. We identified problems that need to be addressed to make traceability more beneficial for software practitioners. An important finding of that study was that practitioners struggle to implement inter-organizational traceability workflows. The interviewed practitioners demanded for guidance on how to enable the usage of traceability across organizational boundaries. Similarly to our findings,

other researchers stated that outsourcing complicates requirements traceability [8, 4]. Although, the problem of inter-organizational traceability was recognized by these researchers, none of them studied it or provided possible solutions.

Lormans et al. [9] conducted an industrial case study with a single international company to understand how requirements traceability is impacted by the outsourcing context. The authors specified requirements for a requirements management system in the context of inter-organizational development. They used these requirements to customize a commercial requirements engineering tool to the needs of their industrial partner. The proposed approach is valuable for the studied case. However, the authors did not focus on a more general understanding of inter-organizational projects as well as the application of requirements traceability in that context. Damian and Chisan [10] studied inter-organizational project relationships in general and identified mistrust and power struggles as critical issues. The authors mentioned requirements traceability as a possible solution to overcome these issues. Alvare et al. [11] studied factors that shape client-supplier relationships and their impact on food traceability. Similar to our approach, the authors propose a conceptual framework of relationships between actors in a distributed production environment.

As a result of our review of related work we can conclude that IT outsourcing is a commonly applied strategy pattern and of high relevance to the software industry. Further, various researchers recognized that inter-organizational specifics need to be carefully considered when applying requirements traceability in the outsourcing context. Beside the small case study on requirements traceability within an international company by Lormans et al. [9], there are no empirical studies that examined the specific challenges of leverage requirements traceability in inter-organizational projects. To close this gap in this relevant area of software development, we found that a systematic study of this problem is necessary.

3 A Taxonomy of Inter-organizational Software Projects

Over the past years, several studies reported a general shift of paradigm from static functional organizations to organizations composed of rapidly changing temporary projects, often referred to as "projectification" [12]. The following rationales for advancing this paradigmatic change emerged as most important. (I) Flexible project organizations allow task-specific resource allocation and avoid long-term resource commitments [13]. (II) Well informed consumers in globalized markets are demanding genuinely innovative products with reasonable pricing and quality that satisfy varying needs [12].

In parallel with projectification, many companies concerned with software development followed IT outsourcing strategies, which we call inter-organizational projectification. The following rationales for this inter-organizational projectification were identified. The fact that well educated people are scarce resources led to high labor costs or even worse to the inability to develop the software product. Outsourcing provided the ability to capitalize a globalized resource pool and to

address the scarce resource issue [2]. DiRomualdo and Gurbaxani [3] synthesized the more general strategic intent of IT outsourcing to leverage external IT assets such as applications, operations, infrastructure, and know-how.

We decided to study this problem in more detail due to the practical relevance of IT outsourcing on the one hand ("inter-organizational projectification") and the limited empirical knowledge on how to deal with requirements traceability in outsourcing scenarios on the other hand. Our research is motivated by the following research questions:

Q1: Who are key actors in inter-organizational projects?
Q2: How do actors interact in inter-organizational projects?
Q3: What goals do interacting parties have in inter-organizational projects?
Q4: How can traceability be leveraged to accomplish Q3's objectives?

Guided by our research questions, we developed a taxonomy that conceptualizes relationships of actors in inter-organizational projects. These actors were characterized by their goals (see Section 3.1). Different interactions between these actors were then described in detail (see Section 3.2).

3.1 Actors and Their Perspectives

According to the definition of Jones and Lichtenstein [14] *inter-organizational projects involve two or more organizational actors from distinct organizations working jointly to create a tangible product/service in a limited period of time. These actors minimally refer to a client and a contractor.* Based on this definition, we identified two types of actors: **client** and **supplier**.

Both, client and supplier simultaneously cooperate in temporary projects and are embedded in their own organizational context [15] as illustrated in Figure 1-a. That means that every actor in an inter-organizational project has an organizational and a project perspective. As both perspectives need to be satisfied, each actor pursue two types of goals, a strategic and a tactical goal. *Strategic goals* describe objectives from an organizational perspective. *Tactical goals* describe objectives from a project perspective (see Figure 1-b). On the one hand, this implies that project team members of client and supplier share the common tactical project goal to develop a certain software product within distinct time and budget [15]. On the other hand, client and supplier organizations also have their individual strategic goals. With strategic goals we mean business goals such as efficiency, innovation, and risk management [16] that are pursued by companies to ensure competitiveness and profitability.

3.2 Interactions

Interactions between client and supplier as well as organization and project are required to align various goals [17]. Based on these often conflicting goals, actors need to interact inter- and intra-organizationally. Figure 2 summarizes these interactions as a taxonomy.

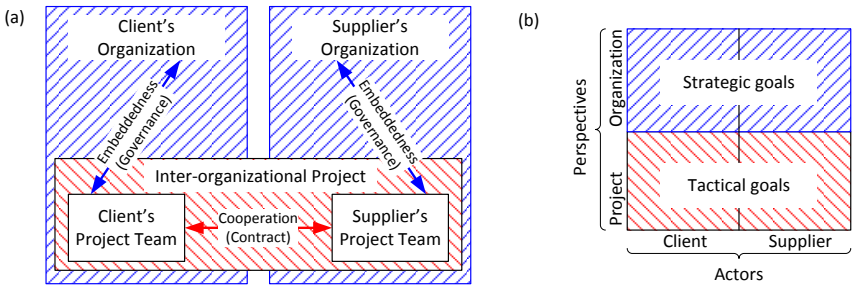

Fig. 1. Actors and their perspectives in inter-organizational projects

Inter-organizational interactions emerge from the cooperation relation between client and supplier. A client-supplier relationship is grounded on contracts. Contracts between client and supplier define exchanges of service and/or products, financial matters, service enforcement and monitoring methods, and communication and/or information exchanges[17, 18]. Thus, we distinguish four different inter-organizational interactions (see Figure 2-a) within our taxonomy: (I) transfer product & service, (II) monetary compensation (III) enforce & monitor quality & progress, and (IV) communicate with project partner.

Intra-organizational interactions emerge from the embeddedness of projects in an organizational context. This embeddedness relation is determined by the fact that tactical project goals need to be aligned with strategic organizational goals. The concept of organizational alignment is known as IT governance. As outlined in Figure 2-b, IT governance can be divided into three types of intra-organizational interactions: (V) compliance verification, (VI) strategic alignment, and (VII) operational excellence [19].

In the next section (see Section 4), we are using the developed taxonomy as a framework to conduct an interview study. We interviewed software development experts from various companies for this study. The findings of our study are then structured according to the developed taxonomy (see Section 5).

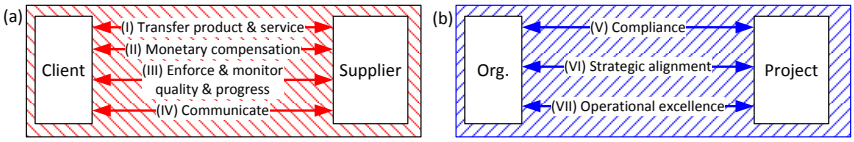

Fig. 2. Overview of (a) inter-organizational and (b) intra-organizational interactions within distributed projects

4 Interview Study

The objective of our study was to gain a better understanding of traceability workflows across organizational boundaries. Thus, we conducted interviews with informants from 17 different companies. Every informant was interviewed in an individual session to prevent that different informant's opinions interfered with each other during the interview. We chose this qualitative research approach for the following reasons. Workflows in inter-organizational projects are complex and multi-faceted. Thus, it would be difficult to define specific context variables required for a quantitative research methodology. In addition, our qualitative interview approach ensured that we were close to the studied software projects and its participants. This proximity helped us to gained an in-depth understanding of the mechanics behind the observed phenomena and avoided misinterpretations during the qualitative content analysis.

4.1 Sampling Cases for the Study

We assembled a list of potential companies from the membership list of the association of friends of the Technical University Ilmenau. This list was extended by contacts we made at a practitioners forum on requirements engineering. We considered every company in the resulting list of 85 companies as a potential case for our study. In order to prioritize this list, we collected general information about each company and identified contact persons from the internet. We then developed a case sampling strategy in order to select the most suitable companies and informants for our study. Following the framework of Curtis et al. [20], we defined and used the following sampling criteria:

- How relevant are general case characteristics to our taxonomy?
- What potential to generate rich information provides the case?
- How generalizable are findings from this particular case?
- What resources (e.g., money and time) are required to study this case?
- Does any ethical issues force us to exclude this case from our list?

After prioritizing the list of potential cases, the contact persons of highest prioritized cases were contacted in order to arrange an interview. Provided that the sampled company agreed, we conducted either one or multiple interviews with key informants of this company. Every informant was interviewed in an individual interview session to avoid influences between informants.

4.2 Data Collection

We decided to employ a semi-structured interview technique with closed-ended and open-ended questions. This approach aimed to guarantee that our investigations are guided by theory, while keeping the flexibility to explore unforeseen information. As described in Section 3, we synthesized our taxonomy from literature. While developing our interview questionnaire, this taxonomy served us as theoretical guidance. The questionnaire consisted of three parts:

1. **General company and project characteristics:** we collected background information about the key informant and the company. Then we asked the informant to describe the software development project she or he is currently involved or has recently finished.
2. **Software development process:** we asked for important process elements such as activities, tasks, roles, stakeholders, artifacts, and tools. Thereby, we aimed to generate a holistic view on the software development process from the beginning to the end.
3. **Inter-organizational traceability workflows:** we asked the informant to provide us with her or his definition of requirements traceability. The answer to that introductory question enabled us to subsequently verify that informants from different cases shared a common understanding of that concept. We then collected characteristics of requirements traceability workflows across organizational boundaries. Hence, we asked for requirements traceability objectives and challenges.

We applied a two phased approach for conducting the interviews. (I) We selected a company in close proximity and performed a three hours lasting interview, which we considered as a pilot run. In result, we produced interview minutes and field notes. We analyzed the interview minutes in order to reveal and eliminate conceptual weaknesses from the questionnaire. We further conducted a retrospective review of our field notes to improve our interview tactics. Thereby, we realized the necessity to approach certain topics differently in order to avoid unwittingly influencing the informant. (II) The actual interviews were conducted with 20 informants from 17 different companies. All interviews were recorded in writing by a designated minute taker.

4.3 Data Analysis

To extract findings from the written interview minutes we applied qualitative content analysis [21]. Our taxonomy served as a qualitative description model. We derived a system of codes from our taxonomy. We classified informant's statements of all written interview minutes and field notes using the defined codes and the qualitative analysis tool MAXQDA10[2].

4.4 Data Demographics

In our study a single case referred to a distinct company concerned with software development. Table 1 outlines that our study contains *small* (less than 100 employees), *medium* (100 to 1,000 employees), *large* (1,001 to 10,000 employees), and *huge* (more than 10,000 employees) companies. It can be seen that small and medium companies mainly conduct *small* (less than 5 project member) or *medium* (5 to 9 project member) projects, while large and huge companies mainly conduct *large* (10 to 100 project members) or *huge* (more than 100

[2] http://www.maxqda.com

project members) projects. Most companies (15) are headquartered in Germany. The remaining companies two headquarters are USA and Austria. All projects were spread across multiple locations, often across multiple countries such as Germany, USA, India, Bulgaria, Czech Republic, Austria, France, and Croatia. The studied companies are active in various domains (Avionic, Finance, Insurance, Logistics, Retail, Security, Transportation) and produce different offerings (Software Product, Hardware Product, Software Development Services). The captured projects represented various types of software development projects, namely New-development, Maintenance, and Migration.

Table 2 outlines characteristics of the interviewed informants. To provide more context to the reader, the table relates every informant to the case she/he belongs to. The table shows that our data covers both actors (Client and Supplier) and both perspectives (Project and Organization). Informant's primary roles are spread across all phases of the software development process (Project Management, Requirements Analysis, Implementation, and Verification).

Table 1. Characteristics of studied companies and their projects

Project size	**Company size**	**Case**	**Project type**	**Offering**	**Domain**
< 5	< 100	**I**	New-development	Service	Public service
		IV	New-development	SW Product	Retail
		VI	Maintenance	HW Product	Robotic
	100..1,000	**VIII**	Maintenance	SW Product	Finance
5..9	< 100	**II**	Maintenance	Service	Insurance
		XI	Maintenance	Service	Insurance
	100..1,000	**VII**	Maintenance	Service	Finance
		XIII	New-development	Service	Finance
	$> 10,000$	**X**	New-development	Service	Insurance
		XII	Maintenance	Service	Retail
10..100	< 100	**XVI**	Maintenance	SW Product	RE
	1,001..10,000	**IX**	Maintenance	SW Product	IT Security
	$> 10,000$	**XIV**	New-development	HW Product	Avionic
> 100	100..1,000	**XVII**	Maintenance	HW Product	Telecommunic.
		V	Maintenance	SW Product	Retail
	1,001..10,000	**III**	Migration	Service	Finance
	$> 10,000$	**XV**	Migration	Service	Logistic

Table 2. Inter-organizational perspective and primary role of interviewed informants

Actor	Perspective	Case	Informant	Informant's primary role
Client	Project	III	Inf-III-1	Project manager
		XV	Inf-XV-1	Business analyst
	Organization	X	Inf-X-1	Process manager
		XII	Inf-XII-1	Portfolio manager
Supplier	Project	I	Inf-I-1	Developer
		I	Inf-I-2	Project manager
		II	Inf-II-1	Project manager
		IV	Inf-IV-1	Development lead
		V	Inf-V-1	Development lead
		VI	Inf-VI-1	Development lead
		VII	Inf-VII-1	Specification manager
		VIII	Inf-VIII-1	Project manager
		IX	Inf-IX-1	Development lead
		X	Inf-X-2	Release & Configuration manager
		XI	Inf-XI-1	Development lead
		XII	Inf-XII-2	Test manager
		XIV	Inf-XIV-1	Tester
		XVI	Inf-XVI-1	Development lead
	Organization	XIII	Inf-XIII-1	GRC manager
		XVII	Inf-XVII-1	Process manager

5 Study Results

This Section provides insights on the results that we extracted from the interview minutes and field notes. Following our taxonomy, we present our extracted findings from different viewpoints. We discuss the client's viewpoint in Section 5.1, supplier's viewpoint in Section 5.2, and the organizational viewpoint in Section 5.3 on requirements traceability in inter-organizational outsourcing projects.

5.1 Client's Viewpoint

In this section we report findings that are specific to the client of an inter-organizational relationship. These findings are structured according to the interaction types of our taxonomy.

Transfer product and/or service: The quality of the delivered end-product was mentioned as most important by all informants. *The supplier hands over a fully verified roll-out baseline to us [Inf-III-1], we insist on a proof of full requirements coverage from the supplier [Inf-XV-1].* Due to contractual obligations, the

client demanded a proof of quality via traceability from the supplier that can be objectively assessed. Especially, clients in strictly regulated environments referred to requirements traceability as a must. Though, traceability appeared to be of great support to objectively assess product quality, two main issues were reported by clients. (I) First, differences in tooling, methodology, and processes between client and supplier made it difficult to efficiently leverage requirements traceability. Main reason for this gap is the fact that technology and processes of each organization were primarily aligned to the organizational goal. That implies that traceability can typically only be used efficiently if this gap is bridged. (II) Second, the existence of one common project goal and two independent organizational goals implied a natural conflict. As a result, traceability information could not or only partially be used across organizational boundaries as its complete disclosure would contradict with supplier's organizational goals.

Monetary compensation: *Change and executive board of the client formally accept and release the roll-out baseline. The final payment is made when this critical milestone is reached [Inf-III-1].* Clients typically defined quality gates that needed to be passed before any kind of payment was executed to the supplier. The assessment of whether or not a quality gate had been passed is a very difficult task for the client. Typically, the client had no direct access to resources at the supplier's side that would be able to provide required input for this assessment. In this case, traceability was the only source that could be used by clients for assessment. Client informants highlighted the issue that traceability information must be reliable due to its high financial impact.

Enforcement and monitoring: *Traceability is used by the client's project managers to control the supplier's progress and quality [Inf-XV-1, Inf-XII-1].* Primary task of the client's project managers was to continuously monitor whether or not the project can still be finished in time and budget and with the expected result. As the client's project managers had typically no direct access to all resources at supplier's side, they required access to reliable traceability information that could be used to measure project progress properly. *All test cases created by the supplier must be accepted and released by the client side before any test execution activity can be started [Inf-III-1].* For a complex scenario with multiple suppliers, the client's project manager pointed out that all supplier activities were synchronized with the help of traceability.

Communication: *Traceability information prepared by the supplier provides valuable input for our further release planning [Inf-XV-1].* Due to the fact that the supplier developed the software, product specific knowledge was generated by the supplier's team members. This product specific knowledge provided valuable input for the client's product manager. The limited access to the supplier's resources forced the client's product manager to gain product specific knowledge indirectly via traceability information. Nevertheless, the supplier's organizational goal of keeping technical or functional knowledge confidential often contradicted the goals of the client's product manager.

5.2 Supplier's viewpoint

In this section we report findings that are specific to the supplier of an inter-organizational relationship. These findings are structured according to the interaction types of our taxonomy.

Transfer product and/or service: *We use traceability to proof the completeness of our implementation to the client [Inf-XIV-1].* The supplier needed to proof the quality and completeness of the implementation in order to avoid expensive disputes. Client and supplier often contractually agreed upon penalties for the case that the delivery of a product with a certain quality was missed. *Our client issues a bug in the application. In case of a false alarm (no bug present) we leverage traceability to proof that the system works as specified by the client [Inf-I-1, Inf-VII-1].* Usually, repairing software defects is covered by the supplier's warranty. Many suppliers reported on the common scenario that a client raised a bug by mistake even though the software was working as specified by the client. Due to warranty obligations, the supplier had either to proof that the product is working properly or to fix the bug. Without traceability between client's requirements and supplier's implementation/verification artifacts the proof of correctness was almost impossible.

Monetary compensation: *When disputing with our clients about product reliability, we use traceability to proof that we did not act with gross negligence in order to avoid paying punitive damages [Inf-XVII-1].* Software errors may have extraordinary impact. In such cases the supplier must be able to proof that she or he did not act with gross negligence. Otherwise, the client may demand compensation, which could even threaten the supplier's existence.

Enforcement and monitoring: *We leverage traceability to monitor our progress and communicate reliable release dates to our clients [Inf-IX-1].* The supplier's project manager used traceability to track the project progress. This information was important to estimate and communicate reliable release dates. Additionally, the project could be monitored to predict project delay. *We must have traceability information to successfully pass quality audits, which are periodically operated by our clients [Inf-XVII-1].* Two informants reported on the fact that they were forced by the client to provide traceability. Otherwise, the client would not even consider entering into a contractual relationship with the supplier. The client regularly verified traceability by supplier audits. Especially, informants working in highly regulated domains highlighted this issue.

Communication: *Our product serves the needs of three different client types. When writing technical product specifications, we typically trace back to the origin of requirements in order to really understand the specific need [Inf-V-1].* Technical project team members at the supplier's side such as designers, architects, developers, or testers directly or indirectly depended on a proper understanding what software was supposed to be built. To gain this understanding a direct communication with the client's requirements engineers was required. Though, direct

communication was limited due to organizational boundaries. Thus, traceability was used to reduce the necessity for direct communication.

5.3 Organization's Viewpoint

The following findings were extracted from interviews with informants that represents the organizational perspective.

Compliance: *Internal auditors reproduce executed software development processes of critical projects [Inf-XVII-1].* Suppliers that developed software for regulated markets were obligated by legal regulation. Thus, compliance with legal regulation was a strategic company goal. Retrospective audits of the software process were supposed to verify whether or not project execution conformed with regulation. Traceability was required to reproduce the process.

Strategic alignment: *We monitor aggregated traceability information from all projects across the company to identify bug hot spots [Inf-XIII-1].* The purpose of identifying bug hot spots with traceability is twofold. First, the risk of delivering low quality products had to be reduced for the company. Second, bug hot spots indicated that the used technology or architecture in this area was insufficient.

Operational excellence: *We use requirements traceability information to establish an early warning system for predicting project crashes [Inf-XVII-1].* Traceability information of running projects could be compared with traceability information of previous projects. By this comparison, critical project evolution could be identified and counter-measures were taken.

5.4 A Practicioner's Checklist

As a conclusion of our previously discussed findings, we derived three success criteria for requirements traceability in inter-organizational projects. We substantiated each success criteria with a list of questions that can be used by practitioners as a checklist.

Criteria I: Ensure availability and reliability of traceability

- What traceability information is required from our project partner?
- Do we rely on traceability information provided by the project partner?
- Is the provision of traceability information contractually specified?
- How can we assess our project partner's trace recording process?
- What are our traceability information quality gates?

Criteria II: Identify and mitigate conflicting objectives

- Do we understand our project partner's organization sufficiently to identify conflicting objectives?

– Are there any conflicting objectives that discourage our project partner from providing necessary traceability information?
– How to establish trust between client and supplier to mitigate conflicting goals?
– Do we need measures to mediate conflicting objectives (e.g. signing non-disclosure agreement)?

Criteria III: Bridge the technological gap between client and supplier

– How does our project partner provide traceability information?
– Are we able to effectively use provided traceability information?

While the above checklist provides guidance for practitioners planning inter-organizational projects towards addressing potential traceability problems, further research effort must be devoted to the question how to address each of the discovered and discussed problems (see Sections 5.1, 5.2, and 5.3). We found that due to the organizational boundary and often contradicting organizational strategies, applying requirements traceability becomes more complicated. We would like to encourage the research community to seek for appropriate traceability solutions for inter-organizational project setups.

6 Threats to Validity

When planning and conducting our study we carefully considered validity concerns. This section discusses how we mitigated threats to the validity.

6.1 External Validity

Due to their nature, interview studies cannot be replicated as identical interview circumstances cannot be recreated. Qualitative studies are primarily concerned with describing and understanding existing phenomena. We described such observed phenomena from our interviews. In an attempt to make these findings usable to other practitioners, we developed a list of critical success factors for requirements traceability in distributed project (see Section 5.4). The fact that our cases diverge across multiple domains, locations, and sizes contributes to the applicability of our findings. However, we are aware of the fact that this kind of study is not generalizable.

6.2 Internal Validity

The instrumentation threat was addressed by applying qualitative content analysis [21], which must be guided by theory from the beginning. We derived a taxonomy from literature as described below. Activities of our study, like creating the questionnaire, conducting the interviews, and extracting the findings were all guided by this taxonomy. We mitigated the threat of case selection bias by defining the selection scheme described in advance (see Section 4.2).

6.3 Construct Validity

Our study is grounded on a taxonomy that conceptualizes inter-organizational software development projects. Thus, we describe and justify how our taxonomy was constructed. To determine the number of potentially relevant studies, we conducted a preliminary search for existing meta-studies on our topic. Then, we extracted primary (P: *inter-organizational*) and secondary (S: *software project*) search term categories from our research questions. Then, we extracted synonym keywords for both categories from labels (headlines, captions) of the evaluated meta-studies as additional search terms (see Table 3). The cross product of both search term categories ($P \times S$) defines our super set of 16 combined search terms. Searching with all these 16 terms produced a list of 9157 unique hits. We narrowed this list to 76 hits by applying the following inclusion criteria: (i) a publication's title must contain a primary search term (ii) a publication's abstract must contain a secondary search term. We carefully read and compared the abstracts of the remaining publications. Additionally, we studied the abstracts of their referencing and referred publications. Following this procedure, we found publications that present definitions on inter-organizational projects as well as typical client/supplier interactions. We then created a taxonomy that is synthesized from overlapping information across these publications. We consider the created taxonomy a reasonable framework for our study as the process for selecting publications followed in principle the accepted Kitchenham method and the publications that we built upon are well referenced by other researchers.

Table 3. Categorized search terms for the literature study

Primary terms (P)	**Secondary terms (S)**
inter-organizational	*software project*
cross-organizational	*software development project*
outsourcing	*software process*
	software development process
	software workflow
	software development workflow

6.4 Conclusion Validity

As described in Section 4.2 we employed a preliminary prototypical interview under realistic conditions to improve our questionnaire and our questioning technique. Thereby we emphasized on eliminating influencing information from questions or questioning behavior. All interviews of our study were conducted with one informant in a single session without break. We offered no room for distractions and interruptions during the interview in order to avoid influences on subjects' answers.

7 Conclusions and Future Work

In this paper we focused on the characteristics of inter-organizational software development projects. In particular, we were interested in whether requirements traceability can help to overcome challenges that a distributed development inherits. We identified three problem areas as most challenging for practitioners.

Different **organizational background** from client and supplier pose the challenge that different technology and methodology is used, e.g., for requirements engineering and software development. This gap needs to be bridged in order to provide sufficient requirements traceability. Although, adapters and tool chains are implemented to handle that issue, practitioners repeatedly reported on their struggle with this technological gap. Inter-organizational outsourcing projects are of temporary nature and client-supplier relationship are manifold. Thus, requirements traceabiliy must be customized for every project. Requirements traceability should therefore be defined as a strategic goal.

Due to **organizational boundaries** between client and supplier the access to artifacts created by the project partner is typically restricted. Negotiated contracts specify artifacts to be delivered either by the client or the supplier. These deliverables are typically only a very small subset of all artifacts created during the requirements engineering and software development of a system. This reduced set of deliverables is often not sufficient to accomplish comprehensive requirements traceability. That implies that requirements traceability needs to be planned very carefully in advance and access to the necessary artifacts guaranteed via contract.

Conflicting objectives exist in two dimensions. The first dimension of conflict is client vs. supplier objectives. The second dimension of conflict is organizational vs. project objectives. Resolving a conflict in one dimension may negatively impact the other dimension. Therefore, a traceability strategy should address all conflicting objectives. If the demand for certain requirements traceability information is contradicting an objective then this traceability information will likely not be provided by the project partner. Eventually, the challenge is to reach a trade-off mediating all four objectives.

Though, we conducted a broad interview study with cases from various domains, more empirical knowledge is required for a generalizable theory. We plan to extend our study and to iteratively evolve the provided practitioner guidelines on requirements traceability in inter-organizational projects. Additionally, further research on supporting tools and approaches is required to provide solutions in the three problem areas identified above.

Acknowledgment The authors would like to thank all practitioners participating in the interview study. We are supported by the German Research Foundation (DFG): Ph49/8-1 and the German Ministry of Education and Research (BMBF): Grant No. 16V0116.

References

1. Gotel, O.C.Z., Finkelstein, A.C.W.: An analysis of the requirements traceability problem. In: Proc. of the First Int. Conf. on RE, IEEE (1994) 94–101
2. Herbsleb, J., Moitra, D.: Global software development. Software, IEEE **18**(2) (2001) 16–20
3. DiRomualdo, A., Gurbaxani, V.: Strategic intent for it outsourcing. Sloan Management Review **39**(4) (1998) 67–80
4. Cleland-Huang, J., Settimi, R., Romanova, E., Berenbach, B., Clark, S.: Best practices for automated traceability. Computer **40**(6) (2007) 27–35
5. Ramesh, B., Powers, T., Stubbs, C., Edwards, M.: Implementing requirements traceability: a case study. In: 2nd Int. Symposium on RE, IEEE (1995) 89–95
6. Arkley, P., Riddle, S.: Overcoming the traceability benefit problem. In: Proc. 13th IEEE Int'l Conf. on Requirements Engineering, 2005. (2005) 385–389
7. Mäder, P., Gotel, O., Philippow, I.: Motivation matters in the traceability trenches. In: Proc. of 17th Int'l Requirements Engineering Conference (RE09). (2009)
8. Gotel, O.: Contribution Structures for Requirements Traceability. PhD thesis, Imperial Collage of Science, Technology and Medicine, University of London (1995)
9. Lormans, M., van Dijk, H., Van Deursen, A., Nocker, E., de Zeeuw, A.: Managing evolving requirements in an outsourcing context: an industrial experience report. In: 7th Int'l Workshop on Principles of Software Evolution. (2004) 149–158
10. Damian, D., Chisan, J.: An empirical study of the complex relationships between requirements engineering processes and other processes that lead to payoffs in productivity, quality, and risk management. IEEE TSE **32**(7) (2006) 433–453
11. Rábade, L., Alfaro, J.: Buyer–supplier relationship's influence on traceability implementation in the vegetable industry. PSM **12**(1) (2006) 39–50
12. Midler, C.: "projectification" of the firm: the renault case. Scandinavian Journal of Management **11**(4) (1995) 363–375
13. Grabher, G.: Temporary architectures of learning: knowledge governance in project ecologies. Organization studies **25**(9) (2004) 1491–1514
14. Jones, C., Lichtenstein, B.: Temporary inter-organizational projects: How temporal and social embeddedness enhance coordination and manage uncertainty. The Oxford handbook of inter-organizational relations (2008) 231–255
15. Bakker, R., Knoben, J., De Vries, N., Oerlemans, L.: The nature and prevalence of inter-organizational project ventures: Evidence from a large scale field study in the netherlands 2006–2009. IJPM **29**(6) (2011) 781–794
16. Von Krogh, G., Nonaka, I., Aben, M.: Making the most of your company's knowledge: a strategic framework. Long range planning **34**(4) (2001) 421–439
17. Kern, T., Willcocks, L.: Exploring information technology outsourcing relationships: theory and practice. JSIS **9**(4) (2000) 321–350
18. Halvey, J., Melby, B.: Information technology outsourcing transactions: process, strategies, and contracts. Wiley (2005)
19. Van Grembergen, W., De Haes, S., Guldentops, E.: Structures, processes and relational mechanisms for it governance. SITG **2004** (2004) 1–36
20. Curtis, S., Gesler, W., Smith, G., Washburn, S.: Approaches to sampling and case selection in qualitative research: examples in the geography of health. Social Science & Medicine **50**(7) (2000) 1001–1014
21. Mayring, P.: Qualitative content analysis. In: Forum Qualitative Sozialforschung/-Forum: Qualitative Social Research. Volume 1. (2000)