Alignment Practices Affect Distances in Software Development: A Theory and a Model

Bjarnason, Elizabeth; Smolander, Kari; Engström, Emelie; Runeson, Per

Published in:
GTSE 2014 Proceedings of the 3rd SEMAT Workshop on General Theories of Software Engineering

DOI:
10.1145/2593752.2593757

2014

Alignment Practices Affect Distances in Software Development: A Theory and a Model

Elizabeth Bjarnason[1], Kari Smolander[2], Emelie Engström[1], Per Runeson[1]

[1]Dept. of Computer Science
Lund University
Lund, Sweden
{ elizabeth | emelie.engstrom | per.runeson }@cs.lth.se

[2]Lappeenranta University of Technology
Software Engineering and Information Management
Lappeenranta, Finland
kari.smolander@lut.fi

ABSTRACT

Coordinating a software project across distances is challenging. Even without geographical and time zone distances, other distances within a project can cause communication gaps. For example, organisational and cognitive distances between product owners and development-near roles such as developers and testers can lead to weak alignment of the software and the business requirements. Applying good software development practices, known to enhance alignment, can alleviate these challenges. We present a theoretical model called the Gap Model of how alignment practices affect different types of distances. This model has been inductively generated from empirical data. We also present an initial version of a theory based on this model that explains, at a general level, how practices affect communication within a project by impacting distances between people, activities and artefacts. The presented results provide a basis for further research and can be used by software organisations to improve on software practice.

Categories and Subject Descriptors

D.2.9 [**Management**], D.2.1 [**Requirements/Specifications**], D.2.4 [**Software/Program Verification**]

General Terms

Theory, Human Factors, Documentation

Keywords

empirical software engineering, software development, distances

1. INTRODUCTION

Coordination and communication within software development [2] is affected by distances [3]. The effects of geographical, socio-cultural and temporal distances are fairly well known and researched for globally distributed software development (GSD) [1][8]. However, the role of distances within co-located development projects and teams has not been explored to the same degree even though there are indications that other types of distances, e.g. organisational, cognitive and psychological distances also affect how requirements are negotiated, communicated and coordinated [10][19]. The flow of information within an organisation can be improved by mapping and optimizing communication paths [9][15]. However, how the communication along these paths is affected by distances is yet to be explored and understood in more depth.

GTSE'14, June 2, 2014, Hyderabad, India.

In a previous study, we identified a number of practices for aligning requirements engineering and testing (RET) [4]. This work gave rise to the research question of this paper, namely *How are RET alignment practices related to distances?* This question is answered herein by a theory that alignment practices support alignment by affecting various distances between people, between activities and between artefacts in a positive way. In addition, we provide a theoretical model that describes how specific RET alignment practices affect different types of distances. The results can be used by practitioners in pinpointing particularly troublesome distances and alignment practices suitable for addressing these. This new knowledge also provides a basis for further research, e.g. in methods and techniques for supporting project coordination and process improvement.

The presented Theory of Distances was deducted through an iterative analysis of interview data from a case company based on an initial hypothesis. Namely that distance is an underlying factor that may explain the occurrence of RET challenges as well as why certain practices improve alignment. This hypothesis was explored and compared against the interview data resulting in a theoretical model and a theory. The generation and description of the model and the theory was inspired by guidelines provided by Seaman [13] and by Sjøberg et al. [14].

The research underpinning the presented theory is described in Section 2, while the case company is presented in Section 3. Section 4 outlines the research method used to generate our theory. The Theory of Distances is presented in Section 5, while the underlying theoretical model, i.e. the Gap Model, is described in Section 6. We evaluate our theory in Section 7, and then summarise this paper and outline future work in Section 8.

2. BAKGROUND AND UNDERPINNING RESEARCH

The work presented in this paper is based on and combines two areas of research, namely alignment of requirements engineering (RE) and testing, and the role of distances in software development.

2.1 RET Alignment

Aligning, coordinating and avoiding gaps between RE and testing is a challenge within software development projects. This challenge relates to a wide range of issues including organization, process, people, tools, requirement changes, traceability and measurements [4][12]. Practices applied in industry to address these challenges include traceability, model-based approaches and increased communication, e.g. by involving testers early in the project and in requirement reviews [16]. Similarly, Marczak et al.

[9] found that in requirements-driven collaboration there is often close communication between requirements and testing roles; key roles which when absent cause disruptions within the development team.

We previously investigated RET alignment through a large interview study at six development companies [4]. The results include a framework consisting of 10 main challenges and 10 categories of practices (in total 27 practices). Examples of RET challenges include aligning goals within an organisation, requirement specification quality, maintaining alignment during changes, outsourcing etc. The RET practices cover a wide range of areas. The main categories of practices are the intersecting areas, i.e. RE and testing including validation and verification, as well as, change management, tracing, tracing practices, tools, metrics and other practices. The RET study also provided a mapping between challenges and practices, i.e. the challenges found to be addressed by each practice. The RET study identified four high-level factors that greatly affect RET alignment. These factors are the human aspects of development, the quality of requirements, the size of the development, and the incentives for implementing alignment. The human side of software development including communication and coordination between people was found to be vital for alignment in general, so also between requirements engineers and testers. Further, the quality and accuracy of the requirements was found to be a crucial starting point for testing the produced software in-line with the defined and agreed requirements. In addition, the size of the development organisation and its projects is a key variation factor that affects both which challenges that are faced and which tools and practices are suitable for the specific company, size and domain. Finally, the incentive for applying alignment practices such as good requirements documentation and tracing vary. For companies with safety-critical development this incentive is externally motivated, while the motivation is purely internal for non-safety critical cases. This internal motivation for RET practices is often weak due to low awareness of the cost vs. benefit of RET alignment.

2.2 Distance in Software Development

We have previously presented a framework of distances within software development, in particular related to requirements engineering [3]. Through a systematic mapping study thirteen types of distances were identified. Eight of these are people-related distances, while five of them are related to distances within or towards artefacts. The people-related distances are geographical (physical distances), socio-cultural (differences in cultural and social values and normative practices), temporal (dislocation in time caused by e.g. time zones), organisational (differing goals and priorities between units), psychological (perceived effort to communicate with another), cognitive (difference in knowledge and awareness), opinions (difference in views and opinions) and power (culturally accepted differences in power, e.g. between manager and staff). The artefact-related distances are semantic (difference in meaning, e.g. of related specifications), syntactic (dissimilarity of structure, e.g. of specifications), similarity (similarity e.g. between projects), impact (e.g. size of change required to modify a requirement) and adherence (difference between formal documentation and actual enactment, e.g. for software process).

The study [3] also provides an overview of the areas for which distances have been researched so far and identifies areas for which further research is needed. The most mature research was found on distances between people, i.e. geographical, temporal and socio-cultural [1]. These types of distance have primarily been researched within the context of global software development (GSD). Even so, there are reported findings from GSD projects where communication was equally strong or even improved compared to co-located development [6] [17]. These contradicting results indicate that there are additional undiscovered factors at play. Furthermore, our mapping study [3] revealed that distances between related software development areas, e.g. between requirements and testing, are largely un-researched both concerning the interaction between people and between artefacts [3]. This is so despite the potential for using the concept of distance to measure and evaluate the coverage and consistency between related artefacts, such as requirements specifications and test cases. Further research into these areas could thus provide valuable insight into these factors and contribute to improved and optimized software development practices including RE methods and practices for eliciting, negotiating and communicating requirements that would better serve their purpose in development activities.

3. THE CASE COMPANY

The theory generation presented in this paper is based on three interviews at one of the case companies (Company A) of our previous RET alignment study [4] (see summary in Section 2.1). This company develops computer network equipment consisting of both hardware and software. The software development unit (which is the part of the company covered by the interview study) has around 150 employees and applies an iterative development model. A typical software project has a lead time of 6-18 months, around 10 co-located members and approximately 100 requirements and 1,000 test cases. A market-driven requirement engineering process is applied. The quality focus for the software is on availability, performance and security. Furthermore, the company applies a product-line approach and uses open-source software in their development.

Three people were interviewed at Company A. Namely a test engineer and a product manager who had both worked at the company for more than three years, and a project manager.

4. RESEARCH METHOD

The Gap Model was constructed through an iterative and inductive analysis process and a theory that encapsulates this empirically-based knowledge was defined in a final theory formulation step. An overview of the applied research method including input and output is shown in Figure 1. Previously collected empirical data in the form of transcribed interviews was selected for re-analysis in this study. In addition, the previously derived frameworks of distances [3] and of RET challenges and practices [4] were used as input to the inductive analysis, see Section 2. This set of distances and RET alignment practice were gradually refined throughout the analysis as occurrences of distances and practices were identified in the empirical data.

The output consists of the Gap Model and the Theory of Distances, which are presented in this paper. The Gap Model was constructed through the analysis and describes the impact of each RET practice on distances. A theory was formulated to provide an abstracted view of the knowledge represented in the Gap model. The inductive analysis was performed by the first author and

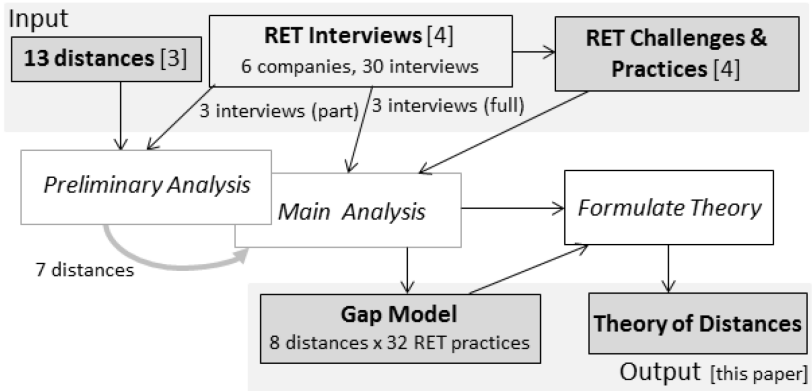

Figure 1. Overview of research method including a) input, i.e. previous work and empirical data, and b) output consisting of the inductively generated theoretical model (Gap Model) and the presented theory, which is an abstraction of it.

partly validated by the third author. The presented theory was formulated by all the authors through repeated discussions.

4.1 Data Selection

No new data collection was performed for the research presented in this paper, rather parts of existing empirical data from our RET alignment study [4] were selected. The choice to re-use this data was guided by the research question defined for this study, namely *How are RET alignment practices related to distances?* The interview data used to previously identify such practices were judged to be a suitable starting point for exploring this question since these interviews explore experienced issues with RET alignment. To further facilitate a thorough but feasible analysis effort, we selected 3 of 30 semi-structured interviews from the RET alignment study, namely those from Company A (see Section 3). We selected this company to represent a typical case by excluding outliers from the full set of case companies. For example, we excluded the consultancy company (Company B) since these interviewees had expert views on a wide range of projects rather than views on projects within one company. The largest company (Company F) was excluded due to the fact that it is far larger than any of the other studied companies. Of the remaining four companies, Company A was the 'average' company concerning size of organisation and projects, and the only company in the set which had pure co-located development. Also, the fact that this company did not have safety-critical development was a factor in favour of selecting Company A. The reasoning for this being that safety-critical development is a significant motivator for applying alignment practices [4]. This indicates that RET alignment within safety-critical development is a special case of software development in general

4.2 Analysis

The selected interview transcripts were analysed in two iterations, namely an initial and a main one as shown in Figure 1. The same analysis process (outlined in Figure 2) was applied in both iterations. The set of distances and RET practices provided as input to the first iteration was gradually refined through this analysis resulting in the distances and practices included in the Gap Model.

The initial analysis iteration was performed on parts of the three interview transcripts. Namely, the parts that relate to the RET challenge *Quality of requirements specification* (Ch3 in [4]) and the practices connected to this challenges. Furthermore, the thirteen distance types derived through our systematic literature study [3] (see summary in Section 2.2) provided the initial set of distances for this iteration. In the initial analysis, we found that seven distances were mentioned in the interview data, namely *abstraction*, *adherence*, *cognitive*, *geographical*, *navigational*, *organisational* and *semantic* distance. Two of these distances were new compared to the original set, namely *navigational* and *abstraction*. *Navigational* distance refers to the effort to navigate between related artefacts. *Abstraction* distance refers to the difference in level of abstraction of artefacts. This set of seven distances was used as input to the next (main) analysis iteration.

We performed the main analysis iteration on the full transcripts of the three selected interviews. The seven distances found in the preceding iteration and the RET challenges and practices reported in our previous study [4] were used as input to this iteration. As in the initial analysis, distances and RET practices were identified in the interview data resulting in distances and practices grounded in the data. The set of distances was further refined, in particular evidence was found for two additional types of distances, namely *psychological* and *temporal* distance, and the *abstraction* distance (from the initial analysis) was included in the semantic distance. This final iteration yielded 8 distances and 32 RET practices (see Section 6.1) for which evidence was found in the empirical data. Furthermore, connections between these practices and distances were identified, also based on the data. The distances and RET practices, and relationship between these are represented in the Gap Model; the output of the analysis. The Gap Model contains eight distances and seven categories of RET practices, in total 32 practices, and knowledge of which distances each practice changes and how, see Section 6.

4.2.1 The Analysis Process

The same analysis process was applied for both the initial and the main iteration of the analysis. The following steps were performed:

- Step 1. Coding of the transcripts
- Step 2. Abstraction and grouping of codes to identify relationships between them
- Step 3. Validation to ensure consistency of the model.

An overview of the analysis process is shown in Figure 2. The input for each iterations consisted of: a set of distances, a set of set challenges and practices, and a set of interview data. The set

of interview data and the set of RET challenges and practices were both extended for main analysis. The input for the two iterations is described in the previous section.

Coding The analysis of the transcripts was focused on identifying distances affecting or being affected by RET challenges and practices. When such information was identified in the transcripts, these parts were coded. A set of initial codes was provided as input to the analysis and consisted of distances and RET alignment challenges and practices. This set was extended and modified during the analysis as additional distances and practices were identified in the transcripts. For example, in the initial analysis iteration the difference in effort to navigate between related parts of artefacts was mentioned by an interviewee. Based on this a new code and thus a new distance type named *navigational distance* was defined to cover this concept.

Abstraction and Grouping In this step relationships between the codes for distances, challenges and practices were identified and abstracted based on the transcripts. These relationships were modelled and visualised in a bi-directional graph. The graph contains distances, RET practices and RET challenges, and relationships between them resulting in triangular relationships (see Figure 2). For example, for the practice of *user/customer testing* (P2.5) relationships were found to cognitive (D4) and adherence (D5) distances, and to (previously shown [4]) RET challenges, e.g. *requirements specification quality* (Ch3). Furthermore, cognitive, semantic, adherence and navigational distances were found to contribute to this RET challenge, which is represented by relationships in the graph.

Through analysis of this representation of the interview data distances relevant for RET alignment were identified. For the initial analysis, the main outcome was this refined set of distances. For the main analysis iteration, the output also consisted of RET practices. Furthermore, for each practice the final output includes which distance the practice addresses and how, i.e. the impact of the practice on a distance.

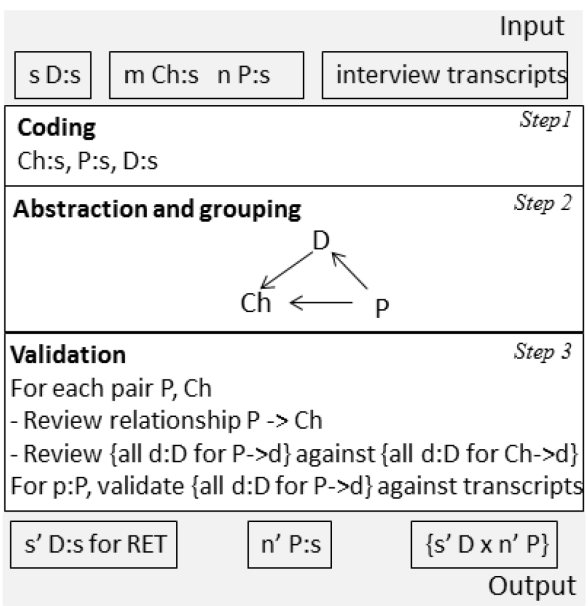

Figure 2. An overview of the analysis process used to refine RE distances (D), RET challenges (Ch), RET practices (P) and connections between these through analysis against interview transcripts.

Validation In the main analysis iteration (but not for the initial one) two researchers performed a validation of a) the analysis and b) its outcome. The first author validated the internal steps of the analysis (i.e. a) with the aim of minimising inconsistencies in the bi-directional graph and in relation to previously published results on RET alignment. The relationships between challenges, practices and distances identified in the analysis and represented in the graph (see Figure 2) were reviewed against the practice-challenge relationships found in the RET alignment study [4]. This was done to ensure internal consistency and to increase the reliability of the Gap Model. This validation was done by comparing the challenge-practice connections in the graph with the ones identified through the RET study. In addition, for each practice addressing a certain challenge, the set of distances contributing to this challenge was reviewed against the set of distances affected by the practices. Ideally these two sets should be the same, although it is possible that additional factors not covered by the identified practices also contribute to a challenge. When discrepancies were identified the relevant parts of the transcripts were re-analysed to resolve the inconsistencies.

The third author validated part of the relationships in the Gap Model by reviewing them directly against the interview transcripts (b). The intent was to mitigate the risk of researcher bias in the identification of distances and the impact of practices on these. Practices were selected by the third author who then searched in the transcripts for evidence of their impact on distances. Discrepancies were resolved through discussions of the contents of the transcripts with the first author (who performed the analysis) and the Gap Model was updated accordingly.

4.3 Formulating the Theory

The details represented in the Gap Model (resulting from the analysis, see previous section) were abstracted and a more general description of the relationship between practices and distances was formulated as a theory. We formulated the Theory of Distances by defining its scope, constructs, propositions and explanations in five basics steps as proposed by Sjøberg et al. [14]. This was done as follows.

- Step 1: The basic constructs of the theory were abstracted from the Gap Model and formally defined.
- Step 2: Propositions were identified by generalising the relationships in the Gap Model and by defining the general relationships between the constructs as identified through the RET alignment study.
- Step 3: The theory was explained at the general level using examples from the empirical data as evidence of the propositions. In addition, a more detailed explanation was provided through a description of the Gap Model.
- Step 4: The scope of the theory was determined by considering case characteristics of the company from which the interview data was taken and the wider set of companies from which the full set of RET practices were derived. In addition, findings from other related empirical studies were compared against our theory in search of supporting evidence.

- Step 5: The validity of the theory's predications has been initially tested with good results by using the Gap Model to identify suitable improvement practices for an agile development team (Paper 6 in [5]). However, this validation of the theory is not part of the scope of this paper.

5. THE THEORY OF DISTANCES

Our theory is here defined and explained. The empirical grounding of the theory is described in Section 6, and the theory is evaluated in Section 7.

5.1 The Formulated Theory

Our theory is presented herein by textual definitions and visualised in Figure 3. We formulated our Theory of Distances through defining a) the scope (or area) for which the theory is put forth, b) the constructs or conceptual elements included in the theory, and c) the propositions or statements of how the constructs are related. The framework presented by Sjøberg et al. [14] has been used as a guideline in articulating our theory.

5.1.1 Scope

The scope within which our theory is of interest is software development organisations and projects. This is a wider scope than the scope of validity [14], namely the scope to which the case study of one company can be generalized, i.e. medium-sized organisations with co-located projects developing embedded non-safety critical software through an iterative development model. Furthermore, the scope of validity is currently the coordination of the areas of RE and testing. The scope of validity can be broadened in the future by further comparison and testing of our theory against data from other case companies and contexts.

5.1.2 Constructs

Software development practices are software development Activities that are conducted recurrently by Actors and that can result in new or changed Artefacts. This may include both organised use of methods and more loosely organised recurring activities that may use individual tools or techniques. For example, the *actor* tester can perform the *activity* of reviewing the *artefacts* test cases against requirements specification resulting in *changes* to the artefact test cases.

Entity is an Actor, Artefact, or Activity in software development.

Distance is a difference in position or level between Entities that requires effort to traverse to accomplish a software development task.

For example, a *distance* between the semantic meaning of the *artefacts* requirements specification and test cases needs to be *traversed* in order to achieve *full test coverage*.

Change of a distance is a decrease or an increase of a distance, or a change in the effect of the distance (called *bridging*). For example, a *change* that *decreases* the *distance* between the semantic meaning of the artefacts requirements specification and test cases results in requiring *less effort* to achieve full test coverage.

5.1.3 Propositions

Pr1 All software development practices change some distances.

Pr2 Long distances require more effort to effectively communicate and coordinate software development.

Pr3 When a distance is decreased or bridged less effort is required to communicate and coordinate between software development entities.

5.2 Explanation of the Theory

The presented theory abstracts the role of practices in software development and in particular how practices affect distances. The objective of these practices is ultimately to facilitate effective coordination between actors, artefacts and activities within a project in an efficient way. By reducing distances, practices decrease the effort needed to communicate effectively thereby improving on the coordination within a project or a development organisation. A reduction of distance may in some cases require certain distances to increase so that others can be reduced. For example, we observed that *Independent testing* (P3.2) caused an increase in organisational distance (see D2 in Section 6.1) but reduced semantic (see D6 in Section 6.1) and cognitive distances (see D4 in Section 6.1). (See Section 6.2.4 for a more detailed description of the impact of this practice.)

The communication between actors is more effective and efficient when distances have been decreased. Direct communication is facilitated by physical proximity (i.e. D1 geographical distance, see Section 6.1), which enables a faster turn-around time. Furthermore, misinterpretations of what people mean are reduced when the teams, roles and individuals have similar knowledge (i.e. shorter cognitive distance, see D4 in Section 6.1), goals and strategies (i.e. shorter organisational distance, see D2 in Section 6.1), and feel closer to each other (i.e. D3 psychological distance, see Section 6.1). In addition, misinterpretations of the meaning of documented information, e.g. requirements, are reduced when the artefacts correspond well to the agreed requirements (i.e. shorter adherence distance, see D5 in Section 6.1).

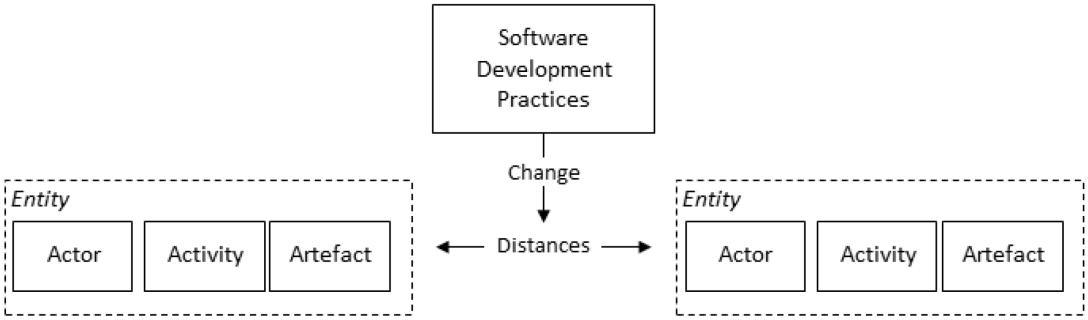

Figure 3. An overview of our Theory of Distances.

When there are long distances between entities, e.g. differences in domain knowledge (i.e. D4 cognitive distance, see Section 6.1), the risk of misinterpreting the information increases, thus requiring more time and effort to communicate and effectively coordinate the software development effort. In some cases, a long distance may also indicate other issues. For example, a long D5 adherence distance between the produced software and the documented requirements may be caused by missed or misinterpreted communication, but also be due to failure to update the requirements specification in a timely manner.

6. GAP MODEL: GROUDNING OF THE THEORY

The developed theory is empirically grounded and the alignment between RE and testing has provided the example area within software development from which we generated this theory. The empirical grounding of the Theory of Distances will now be explained in more detail by a description of the Gap Model from which it was formulated.

6.1 Included Distances and Practices

The Gap Model represents our empirically derived knowledge of the impact of specific RET practices on distances and contains eight distances and seven categories of RET practices. The following distances are defined in the Gap Model.

- **D1 Geographical** The physical distance between the positions of actors' workplaces.
- **D2 Organisational** The distance between actors' placement within an organisational structure, e.g. level within a hierarchy of units and departments.
- **D3 Psychological** The subjective level of effort perceived by one actor to communicate with another, affected by, e.g. personality and opinions.
- **D4 Cognitive** The difference in levels of cognition between actors, e.g. competence levels, knowledge of domain.
- **D5 Adherence** The level of similarity between the contents of an artefact and the actual situation, e.g. the difference between a documented requirement and the actual behaviour of the software.
- **D6 Semantic** The level of similarity in meaning between two related artefacts.
- **D7 Navigational** The difference in position of related parts of different artefact, e.g. path length to navigate between a requirement and the test cases that verify it.
- **D8 Temporal** The distance of the position in time when related activities are performed, e.g. time between defining a requirement and specifying a test case that verifies it.

The RET practices included in the model are listed in Table 1. Four of the categories of practices from the original list [3] are not included due to not being mentioned in the analysed interviews. These non-included categories are (Pn are identifiers used in [4]): Process enforcement (P5), Traceability responsibility roles (P7), Alignment metrics (P9) and Job rotation (P10). Furthermore, 17 additional practices were identified through the analysis performed to construct the Gap Model. These are marked with a + in the full list of included practices provided in Table 1.

6.2 The Impact of Practices on Distances

Each of the included RET practices decrease one or more distances. An overview of the empirically-derived connections including the impact on each distance is shown in Table 2. In some cases the distances are decreased by establishing an alternative communication path that is more efficient than the original one. We call this *bridging* a distance. For example, an organisational distance between requirements engineers and testers can be bridged with the practice *Cross-role requirements reviews* (P1.1). Bringing these roles together in a common meeting and around a common task establishes a direct communication channel between these actors for which there is an organisational distance while this organisational distance remains unchanged. This practice also decreases adherence distance between the agreed and the documented requirements by identifying and resolving these differences between perceived and documented requirements. This decrease of adherence distances reduces later misinterpretations and miscommunications.

To further illustrate the impact of practices on distances, we will now describe seven of the included RET practices and their impact on distances in more detail. This set of example practices have been selected to demonstrate both a simple and a more complex impact of practices on distance. Furthermore, the practices were selected to obtain examples that cover all of the distance types and all but one of the practice categories.

6.2.1 Product Manager Physically Present to Developers and Testers (P1.9)

This practice relates to the physical location of the product manager relative the office space where other roles of a development project have their desks, in particular the developers and the testers. The product owner can either be re-located to a desk closer to this area, or an agreement can be made that the product owner will spend more time at this location.

Impact on Geographical Distance (D1) The geographical distance between a product owner and the developers and testers is decreased by allocating the product owner a desk closer to these roles. Similarly, this distance is decreased, at least part of the time, by the product owner attending project meetings and spending more time in the developers' office space. This increased physical proximity increases the availability of the product owner to the rest of the team. This in turn encourages more frequent and efficient communication of requirements, e.g. clarifications, detecting misunderstandings and conflicts.

Impact on Organisational Distance (D2) This practice can bridge organisational distance between the product owner and the roles to which the physical proximity is increased by providing a more direct communication path between these roles rather than traversing the hierarchical structure of the line organisation.

Impact on Cognitive Distance (D4) Cognitive distances can be bridged and eventually decreased between co-located people. The increased communication and awareness caused by being physically close to each other contributes to sharing knowledge of the domain, process and organisation, and different views on

priorities for the system under development. This increased knowledge share, can bridge cognitive distance in situations where a high degree of domain knowledge is required. For example, it can ensure a common understanding of user requirements between the product owner and a tester. Furthermore, over time the cognitive differences (distance) between these roles can also decrease as the knowledge and perspectives are shared and discussed.

Table 1. RET practices in Gap Model. Practices marked + are added compared to [4]. Practices marked bold are described further in the text.

P1 RE Practices	P1.1 Customer communication at all requirements levels and phases
	P1.2 Development involved in detailing requirements
	P1.3 Cross-role requirements review
	+P1.7 Use of a customer proxy role
	+P1.8 Feature requirements documentation
	+P1.9 Product manager physically present to developers & testers
	+P1.10 Informal communication within organisation
	+P1.11 Product manager involved in development project
	+P1.12 Same process for QRs
	+P1.13 Structure requirements artefacts accord to type
	+P1.14 Collaborative definition of quality reqts
P2 Validation Pract	P2.1 Test cases reviewed against requirements
	P2.3 Product manager reviews prototype
	P2.5 User / Customer testing
	+P2.6 Early test involvement in development projects
P3 Verification Practices	**P3.2 Independent testing**
	P3.3 Testers re-use customer feedback
	+P3.5 Feature-based test plan
	+P3.6 Separate testing team for quality requirements
	+P3.7 Test-impact analysis
	+P3.8 Close cooperation between Test and Development unit and roles
P4 Change	**P4.1 Process for requirements changes involving Test**
	P4.2 Product-line requirements practices
P5 Tracing Practices	P6.1 Document-level traces
	P6.2 Requirements-test case traces
	P6.3 Test cases as requirements
	+P6.5 Conceptual tracing
	+P6.6 Traces between people/roles
P8 Tool Practices	P8.1 Tool support for requirements and testing
	P8.2 Tool support for requirements-test case tracing
+P11 Process	+P11.1 Incremental development
	+P11.2 Small-scale development

Table 2. Overview of Gap Model: empirically-derived effects of RET practices (see Table 1) on distance, B=Bridge, D=Decrease, and I=Increase of distance.

	P1 RE	P2 Validation	P3 Verification	P4 Change	P6 Tracing	P8 Tools	P11 Process & Size
D1 Geographical	D: **1.9**						
D2 Organisational	B: 1.2, 1.3, **1.9**-1.11, **1.14**	B: 2.1, 2.3	I: **3.2**, 3.6	B: **4.1**	B: **6.2**, 6.6		D: **11.2**
D3 Psychological	D: 1.10						D: **11.2**
D4 Cognitive	BD: 1.1-1.3, **1.9**, 1.11 B:1.7, 1.8, **1.14**	B: 2.1, 2.3, **2.5** BD: 2.6	BD: 3.6 B: **3.2**, 3.3	B: **4.1**			D: **11.2** BD: 11.1
D5 Adherence	D: 1.1-1.3, 1.8, 1.10-1.11	D: 2.1, 2.3, **2.5**		D**: 4.1**	D: **6.2**, 6.5 I: 6.3	D: 8.2	D: 11.1
D6 Semantic	D: 1.1, 1.2, 1.8	D: 2.1, 2.3, 2.6	B: 3.6 D: **3.2**, 3.5, 3.7-3.8	D: 4.2	D: 6.1-**6.2**, 6.5	D: 8.2	D: 11.1
D7 Navigational			D: 3.5		D: **6.2**-6.3	D: 8.1, 8.2	
D8 Temporl		D: 2.6		B: **4.1**			D: 11.1

6.2.2 Collaborative Definition of Quality Requirements (P1.14)

Quality requirements, e.g. performance, stability, usability etc, are collaboratively identified. All stakeholders necessary for the relevant quality aspects are involved in defining these requirements.

Impact on Organisational Distance (D2) This practice brings stakeholders and project members together to define quality requirements and, thus provides them with a common objective. Through providing this objective of defining and agreeing to quality requirements the practice can bridge potential organisational distance and provide a forum that shortcuts an organisational hierarchy.

Impact on Cognitive Distance (D4) The priority aspect of cognitive distance can be bridged between project members and stakeholders by bringing these roles together to jointly discuss and identify the quality requirements to aim for in the project. By sharing different views on the priorities for the system cognitive distance between roles can be bridged by understanding the

different perspectives and reaching a common agreement on the quality requirements for a project.

6.2.3 User/Customer Testing (P2.5)

Delivering executable code at regular intervals allows customers and/or end-users to test and validate the product under development, and then provide feedback to the development project.

Impact on Cognitive Distance (D4) This practice can bridge cognitive distance concerning domain knowledge between the customer/user (or a customer proxy) and the developers and testers within the project. Missed or misinterpreted requirements can be identified by utilising the customers' or the users' knowledge of the domain to validate the produced software.

Impact on Adherence Distance (D5) A distance in adherence between the agreed requirements and the produced software can be decreased through this practice. This is achieved by detecting missed or misunderstood requirements and then addressing these, ideally at an early stage in the development cycle.

6.2.4 Independent Testing (P3.2)

This practice entails having a testing team that is separate and independent from the developers, thereby ensuring that testers are not biased by the developers' interpretation of the requirements.

Impact on Organisational Distance (D2) Introducing a separate testing unit will increase the organisational distance between the testers of this team and the roles in the development team. This increased distance will decrease the communication between these testers and the developers (which is the aim of the practice).

Impact on Cognitive Distance (D4) This practice can bridge cognitive distance for the aspects of technical skill in testing and priorities by introducing a testing team focused on testing. The strong competence and focus on testing within this team can then compensate for weaknesses in this area in other project roles.

Impact on Semantic Distance (D6) The semantic distance between requirements and test artefacts for the aspect of coverage can be decreased by this practice. The test competence and clear responsibility for testing prescribed by this practice can lead to improved test coverage of requirements.

6.2.5 Process for Requirements Change Involving Test (P4.1)

Involving testing roles in the decision making and in the communication of changes to the requirements supports alignment through increased communication and coordination of these changes to the test organisation.

Impact on Organisational Distance (D2) Organisational distance between the testers and other project roles, e.g. product owner and developers, is bridged by this practice by introducing direct communication and decision making channels for requirements changes.

Impact on Cognitive Distance (D4) Cognitive distance for the aspect of domain knowledge can be bridged with this practice. This is achieved by ensuring that roles with a high level of domain knowledge are involved in the decision making process for requirements changes.

Impact on Adherence Distance (D5) This practice can decrease adherence distance between delivered software and agreed requirements, both by ensuring that there is a common view on what the agreed requirements are and by initiating changes to the software that brings it closer to what the user has requested.

6.2.6 Requirement-Test Case Traces (P6.2)

Tracing between individual requirements and the test cases that verify these supports a number of activities like impact analysis, ensuring sufficient test coverage and reviewing test cases against requirements.

Impact on Organisational Distance (D2) This practice can bridge organisational distance, e.g. between the roles defining requirements and the testers, by providing direct pointers into relevant entities in the requirements versus test specification. In this way, information is made more readily available to other parts of the organisation and not just to the local unit responsible for maintaining the documentation.

Impact on Adherence Distance (D5) Requirement-test case traces can decrease the adherence distance between agreed and documented requirements. The practice leads to a more active use of the documented requirements (for tracing to test cases) and can thereby also catch differences (distance) in the requirements between what has been agreed and what is documented. This is particularly relevant to a development process where a combination of face-to-face and document-based requirements communication is applied.

Impact on Semantic Distance (D6) The semantic distance between requirements and test cases for the aspect of coverage can be decreased by tracing between the two entities. The tracing simplifies measuring and reviewing the test coverage for requirements to ensure that it is sufficient and that all requirements are tested.

Impact on Navigational Distance (D7) The navigational distance between requirements and test cases is decreased by this practice for the simple reason that the effort required to locate the corresponding entity is decreased. However, the effort required to create and maintain the traces is substantial.

6.2.7 Small-Scale Development (P11.2)

This practice entails organising software development in such a way that it simulates small-scale development. Development is then performed with a small and tight-knit development team, thus avoiding the overhead and complexity related to large organisational structures.

Impact on Organisational Distance (D2) Organisational distance is per default minimised with this practice since organisational structure is consciously removed to construct a small organisation and project.

Impact on Psychological Distance (D3) Psychological distance between individuals can be decreased in small-scale development due to the close day-to-day working relationship within the small development team. However, if there is larger psychological distance between people this may also become more apparent in a small-scale setting, whereas in a larger context these people can avoid each other to a greater extent.

Impact on Cognitive Distance (D4) Cognitive distance between roles and individuals are decreased in a small project. The close collaboration and frequent interaction between people leads to sharing knowledge and perspectives, and learning from each other thereby decreasing cognitive differences or distances.

6.3 Limitations of the Gap Model

We will now report on the known limitations and threats to validity of the Gap Model by discussing the construct, internal and external validity [11] of this part of the study.

Construct validity The choice to use existing interview transcripts poses a risk to construct validity. Since the interviews were not designed to explore the topic of distance, rather the area of RET alignment there is a risk that information on distances is missing and/or that interviewees are incorrectly interpreted to refer to distances. The richness of the interview data partly mitigates this as the original study was designed for a wide and a deep exploration of RET alignment. However, it remains an open risk that not all distances or alternative factors have been identified in the Gap Model. This needs to be addressed by further studies specifically targeted at distances.

Internal validity The main threats to internal validity of the Gap Model are the risk of researcher bias and the risk of failing to systematically and rigorously manage the large amounts of empirical data. As pointed out by Seaman (p. 567 of [13]) since the coding and pattern matching is largely a creative process, there is a risk that the researcher is tempted to rely on hunches and start writing too early, rather than systematically analyse the data and the codes. Furthermore, there is a risk that the prejudice and pre-assumptions influences the analysis process. As for all empirical studies, this remains an open risk despite being partly mitigated by triangulating the outcome of the data analysis against the data. This was achieved by another researcher searching for evidence in the interview transcripts for the relationships for four practices represented in the Gap Model. The set of distances claimed to be affected by the reviewed practices were thus reviewed and the one detected difference was discussed, resolved and the Gap Model was modified accordingly.

External Validity The fact that the Gap Model is currently based on empirical data for one company affects the extent to which it can be generalised. Even so, we believe the model is applicable also to other cases due to having analysed the data from one company against existing frameworks (i.e. distances and RET practices) which have a wider empirical base. In particular, the Gap Model may be valid for organisations and companies with similar contextual characteristics. Namely, for small development projects developing non-safety critical software, in medium sized companies, where all employees are co-located at one site, and for which an iterative or agile development model is applied.

7. EVALUATION OF THE THEORY

Our theory can be classified as a *predicting theory* (type III [7]), i.e. it predicts what will happen when software development practices are applied. Specifically, the Theory of Distances predicts the impact on the distances. The theory also explains at the general level why practices support and improve the process of software development. Compared to an *explanatory theory* (type II [7]) our theory also has testable propositions. We consider that the explanations for the causal relationships, although currently provided (see Section 6) require more detailing before our theory can be classified to be of the type *explanation and prediction* (type IV [7]).

In this section we evaluate the Theory of Distances in software development using the criteria provided by Sjøberg et al. [14]. For each criterion, we rate how well our theory meet is as low, moderate or high.

7.1 Testability

The Theory of Distances is formulated in an understandable and internally consistent way. The constructs and propositions are free from ambiguities, but require a familiarity with the concept of distance. The Gap Model (Section 6) provides such further definitions of distance types and their role in software development. Furthermore, the distances can and have been measured through a combination of physical measurements, surveys and document studies (see Paper VI in [5]).

Hypotheses can be derived from the propositions and then empirically tested. However, due to the wide scope, i.e. software development from large to small development context, makes it is very costly to fully test this theory. In addition, the many and complex interacting factors at play in software development indicate that testing needs to be performed in a real live development context, e.g. through case studies.

We rate the testability of this theory as moderate.

7.2 Empirical Support

There are few studies on the alignment and coordination of RE and testing [16], apart from our previous study [4]. There are more related studies on communication and coordination within software development [9][15] and some on distance, primarily for globally distributed development [1][8]. All of these studies focus on one of the aspects of: RET alignment, communication paths or distances, but not all together and not at the same level of detail as our study. Even so, some empirical support for our theory can be found in the work of others.

The scope of the empirical study by Uusitalo et al. [16] is similar to our RET alignment study [4] but narrower in that it only investigated practices for alignment. Even though the concept of distance was not investigated in these studies, distances can be identified in the descriptions of the found practices. For example, the practice *Early tester participation* improves on the test planning activities both by bringing these activities closer in time to the project planning (changing temporal distance, see D8 in Section 6.1) and by increasing the testers knowledge of the domain and the system (changing cognitive distances towards other actors, see D4 in Section 6.1). The practice *Linking testers with requirements owners* establishes a direct communication path between actors and thereby changing the distance between these and improving on the communication quality and speed. Both of these practices confirm Pr1 and Pr3. An additional aspect of distances currently not covered by our theory is indicated by the practice *Tester participation in requirements reviews* (similar to the practice P1.3, see Table 1), namely that differences, i.e. distance, can be an advantage in identifying problems with the requirements specification. This is an indication that there are additional aspects yet to include in our theory.

The case study by Stapel and Schneider [15] investigated communication within a globally distributed development project. Communication paths were mapped and analysed to find communication issues and suggest practices to improve the information flow. The study does not focus on different types of distances except that the context of the project included geographical and temporal distances. Even so, long distances can be clearly seen to be described in all of the three identified reasons for communication problems, thus providing evidence for proposition Pr2. For example, smooth collaboration within the project was hindered by a group of project members 'not having the same amount of prior context knowledge' (i.e. cognitive

distance, see D4 in Section 6.1). Similar distances or difference (i.e. cognitive) between engineers in 'what they know about each other' was found to have a negative effect on the efficiency of the communication. Information flow problems were observed to be caused by inconsistencies between documented and fluid (undocumented) information, what we call adherence distance (see D5 in Section 6.1).

The case study by Holmstrom et al. [8] investigated the impact of agile practices on distances for a GSD project at Intel. The findings include observations of how certain XP practices reduce geographical, temporal and socio-cultural distances (Pr1) and thereby improve on the communication (Pr3), coordination and control within a GSD project. We interpret their findings to include an impact on cognitive distance (see D4 in Section 6.1) through an increased share of knowledge and to some decrease on psychological distance (see D3 in Section 6.1) in that geographically distributed team members felt more connected.

In summary, we find evidence in other studies [8][15][16] for all of the propositions of our theory. In addition to supporting our propositions the findings from these studies also include additional aspects of distance not covered by the current version of the Theory of Distances. For example, the impact of practice on control within a project [8] and positive effects of differences (distances) in cognition [15]. While supporting our theory, these results also indicate than it is not yet complete. Furthermore, more analysis of related work is needed to provide a fuller view of how existing empirical research relates to our theory.

While our theory is grounded in empirical data, we rate the empirical support for our theory from other studies currently identified as low. Further systematic investigations of the concept of distance in related work are needed to identify more such support.

7.3 Explanatory Power

This theory is defined at a middle range [14], i.e. it is closely linked to observations but involves some abstraction. This initial version of the theory describes the impact of practices on distances in general, while other impacts and motivations for applying practices are not included. Thus, our theory explains some, but not all factors at work within the complex reality of software development.

We judge that the explanatory power of this current version of our theory is low to moderate. Additional work is needed to further explain how different distances interrelate and affect each other.

7.4 Parsimony

Parsimony concerns how economically a theory has been constructed pertaining to the number of constructs and propositions used. Despite the rich and complex view provided by the qualitative data the Theory of Distances was formulated to encapsulate this at a high level. The number of constructs and propositions were kept to a minimum when abstracting the many details of the Gap Model (see Section 6) into our theory.

We consider the parsimony of the theory be high.

7.5 Generality

The scope of the Theory of Distances is defined at the level of software development rather than for RET alignment in general or for the specific case company. The existing frameworks for distances [3] and RET practices [4] on which our theory generation is based represent a wider knowledge base than the one case company used for our inductive analysis. In addition, related empirical studies support the propositions of this theory, see Section 7.2. Also, the results from an initial testing of the Gap Model for a different case company (reported in Paper 6 of [5]) indicate that our theory is generalizable beyond the one case used in the theory generation.

We have used the alignment of RE and testing as an example area within software development for which we have derived a theory of distances within software development. The related studies for communication paths [15] and distances in global software development [1][8] indicate that our theory is generalizable to software development. However, further work is needed to test and validate our theory for a wider range of software development companies and contexts, including other development activities, such as usability design and architecture.

We consider the generality of our theory as high.

7.6 Utility

The presented theory can be used to provide a greater awareness of the importance of communication and the role of distances in enabling or hindering effective communication within an organisation or a development project. This increased awareness of the distances and practices will enable new kinds of approaches to the design and development of software development practices. In addition, the concept of a distance can be used in research. It can open new research opportunities and create new kinds of research instruments.

We consider the utility of the theory as high.

8. SUMMARY AND FUTURE WORK

Coordination and communication are vital factors in the success of a software project and in delivering the required software on time and within budget. Information, e.g. concerning the customer requirements, is often lost or distorted when there are gaps in the communication. These gaps can be caused by distances between people, e.g. organisational and cognitive distances, but also by semantic distances, e.g. between a requirements specification and the test cases created to verify the software. Good practices in software development can improve the communication and bring clarity to different roles, e.g. through cross-role review of requirements, customer testing, product management review of prototypes etc. These industrial practices reduce various distances within a project and thus improve the communication and coordination.

In this paper, we present an initial version of the Theory of Distances that explains how software development practices and distances can affect a software development project. In particular, our theory expresses that practices change distances between actors, activities and artefacts within software development. Furthermore, this paper also presents the underlying theoretical model on which we base our theory, namely the Gap Model. This model describes in more detail which distance types an individual practice affects and how.

Our findings highlight distance as an important factor within software development. The presented knowledge can aid development organisations in identifying and addressing distances contributing to communication and coordination problems. Furthermore, the list of practices included in the Gap Model can be used to identify suitable improvement practices.

Future work includes testing our theory and strengthening its validity for different contexts including size of organisation, domain and range of development activities. This can be achieved by considering evidence from additional cases, both from our own studies and from related studies. Our theory can then be refined through further inductive analysis and the base of empirical knowledge can be extended through empirical testing. In addition, investigations into how different types of distances relate to, and affect, each other and which distances affect which entities both pose an interesting area for future research

9. ACKNOWLEDGEMENTS

This research was partly funded by EASE (ease.cs.lth.se), Ericsson Research and Academy of Finland grant #259454.

10. REFERENCES

[1] Agerfalk, P.J., Fitzgerald, B., Holmstrom Olsson, H., Lings, B., Lundell, B., Conchúir, Ó.E. 2005. A Framework for Considering Opportunities and Threats in Distributed Software Development. In: *Proc. Of Int. Works. on Distr. Softw. Eng*., DiSD 2005, pp. 47-61.

[2] Bjarnason, E., Wnuk, K., Regnell, B. 2011. Requirements are Slipping Through the Gaps – A Case Study on Causes & Effects of Communication Gaps in Large-Scale Software Development. In*: Proc of 19th IEEE International Requirements Engineering Conference,* 2011, pp.37-46.

[3] Bjarnason, E. 2013. Distances between Requirements Engineering and Later Software Development Activities A Systematic Map. In: *Proc. 19th Int. Working Conf. Requirements Engineering: Foundation for Software Quality* (REFSQ'13), pp. 292-307.

[4] Bjarnason, E. Runeson, P., Borg, M. et al. 2013. Challenges and Practices in Aligning Requirements with Verification and validation: A Case Study of Six Companies. *Journal of Empirical Software Engineering*, July 2013.

[5] Bjarnason, E. 2013. *Integrated Requirements Engineering Understanding and Bridging Gaps Within Software Development*. Doctoral Thesis, Dept. of Computer Science, LU-CS-DISS: 2013-02, ISBN: 978-91-7473-732-5, 272 pages. Dept. of Comp. Science, Lund University, November 2013.

[6] Damian D. 2001. An Empirical Study of Requirements Engineering in Distributed Software Projects: Is Distance Negotiation More Effective? In: *Proc of 8th Asia-Pacific Software Engineering Conference* (APSEC 2001), pp. 149-152.

[7] Gregor, S. 2006. The Nature of Theory in Information Systems. *MIS Quarterly* Vol 30. No. 3, pp. 611-642, September 2006.

[8] Holmström, H., Fitzgerald, B., Ågerfalk, P.J., Conchúir, E.O. 2006. Agile Practices Reduce Distance in Global Software Development*. Information Systems Management,* 23:3, pp-7-18

[9] Marczak, S., Damian, D. 2011. How Interaction between Roles Shapes the Communication Structure in Requirements-Driven Collaboration. In: *19th IEEE Int Requirements Engineering Conf.* pp. 47-56.

[10] Prikladnicki, R. 2012 Propinquity in Global Software Engineering: Examining Perceived Distance in Globally Distributed Project Teams. *Journal of Softw. Evolution and Process*, 24(2), pp. 119-137.

[11] Runeson, P., Höst, M., Rainer, A., Regnell, B. 2012 *Case Study Research in Software Engineering Guidelines and Examples.* Wiley.

[12] Sabaliauskaite, G. Loconsole, A, Engstrom, E. et al. 2010. Challenges in Aligning Requirements Engineering and Verification in a Large-Scale Industrial Context. In: *16th Int Working Conf on Requirements Eng. Foundation for Software Quality* (REFSQ), pp. 128-142.

[13] Seaman, C.B. 1999. Qualitative Methods in Empirical Studies of Software Engineering. *IEEE Transactions on Software Engineering*, vol. 25, issue 4, pp 557-572.

[14] Sjøberg, D.I.K., Dybå, T., Anda, B.C.D., Hannay, J.E. 2008 Building Theories in Software Engineering, Chapter 12 in Shull, Singer, Sjøberg (eds.), *Guide to Advanced Empirical Software Engineering*, Springer-Verlag, London.

[15] Stapel, K., Schneider, K. 2012. Managing knowledge on communication and information flow in global software projects. *Journal of Expert Systems*.

[16] Uusitalo, E.J., Komssi, M., Kauppinen, M. et al. 2008 Linking Requirements and Testing in Practice. In: *16th IEEE Int Requirements Eng. Conf*, NJ, USA, 265-270.

[17] Wolf, T., Nguyen, T., Damian, D. 2008 Does Distance Still Matter? *Journal of Impr. and Practice of Softw. Process*, 13(6), pp. 493-510.

[18] Yin, R.K. 2009. *Case Study Research Design and Methods* (4th edition). Sage Publications.

[19] Yu, Y., Sharp, H. 2011. Analyzing Requirements in a Case Study of Pairing. In: *Proc. of 1st Int. Workshop on Agile RE*, Lancaster, UK.