Engineering of Quality Requirements as Perceived by Near-shore Development Centers' Architects in Eastern Europe: the Hole in the Whole

Maya Daneva
Computer Science School
University of Twente
The Netherlands
+31534892889
m.daneva@utwente.nl

Sabrina Marczak
Computer Science School
Pontifícia Universidade Católica do Rio Grande do Sul
Porto Alegre, RS - Brazil
sabrina.marczak@pucrs.br

Andrea Herrmann
Herrmann & Ehrlich
Stuttgart, Germany
+49-711-41006191
herrmann@herrmann-ehrlich.de

ABSTRACT

Context: To software architects (SAs), the quality requirements (QRs) to a software system are key to designing the software architecture. However, understanding SAs' roles in the QRs engineering activities only recently became a topic in empirical requirements engineering research and very little is still known about QRs engineering from SAs' in large and distributed projects. **Goal:** This exploratory study aims at explicating how SAs are involved in engineering QRs in a specific distributed development setting, namely in organizations that distribute software development activities to closely located business units, known as near-shore development centres (NDCs), and in a specific geographic zone, namely Eastern Europe. **Method:** Based on interviews with 16 practitioners working on large projects in NDCs, we explicate the participation and involvement of NDCs' architects in QRs tasks. **Results:** We found that SAs from NDCs (i) are actively involved in QRs documentation and validation, (ii) are relatively passive participants in QRs elicitation and prioritization, and (iii) are not at all involved in QRs negotiation. Perhaps, our most surprising finding is that NDCs may often have economic incentives to misalign with onshore QRs practices. **Conclusions:** We explicated QRs practices, compared them to previously published ones, and found implications for both researchers and practitioners. Though,our results are preliminary, as they are from an exploratory study.

Categories and Subject Descriptors

D.2.1 [**Requirements/Specifications**]: Elicitation methods, Methodologies, Tools.

General Terms

Documentation, Design, Experimentation, Human Factors

Keywords

Quality requirements; Software architecture; Near-shore development centres; Distributed software development; Interview-based research method; Exploratory case study.

ESEM'14, Sept 18-19, 2014, Torino, Italy.

1. INTRODUCTION

Global software development centres and near-shoring have emerged as important research contexts of empirical studies on software business and on software systems delivery [1], but also in the field of requirements engineering (RE). However, empirical research on distributed RE primarily focused on risk identification and risk management [2] and on the application of a broad range of coordination and communication theories to the study of collaboration [3]. Little research so far has been done on specific collaborative RE processes concerning a particular type of requirements, e.g. Quality Requirements (QRs), or a particular RE sub-area, e.g. elicitation, prioritization [3]. QRs are non-functional requirements [28,29], as performance, reliability and usability requirements. E.g. performance requirements describe how fast functionality shall be executed. Specifically, very little has been known about the relationship between engineering of QRs and the distributed development sites involved in downstream software development activities. Yet, QRs are key to determining the complexity of a software system [4], and in turn, dealing with them and trading them off is of paramount importance to project success. According to the V Model XT [30], "the Software Architect is responsible for designing and developing all Software Units and products of the type External Software Module of a System." According to the SWEBOK [29], "a software architecture is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both". Software architects (SA) should feel responsible for QR because they influence architectural decisions: "Often, the impact on quality attributes and trade-offs among competing quality attributes are the basis for design decisions" [29]. As stated in [6], QRs help SAs order their work in both focus and timing of their decisions. The involvement of SAs into engineering QRs, therefore, has recently become a topic of research in empirical RE (e.g.[6,7,8,9,10]). However, because of the recent date of these publications and their limited number, our understanding of the ways in which engineering of QRs varies based on project type, organizational culture, and business model conditions is far from complete [10]. In most of the studies on engineering QRs from SAs's perspective, the assumed context was in-house development while very few studies focused on other systems delivery contexts, e.g. contract-driven and distributed [12]. Responding to calls of other RE researchers [3,9] for more research on QRs processes in specific contexts, in this paper we set out to investigate the perspectives of SAs on QRs in one particular setting, namely

large distributed projects that involve near-shore development centres (NDRs) in Eastern Europe, e.g. in Bulgaria, Romania, Hungary [11]. The term 'nearshoring' means sourcing service work to a foreign country that is relatively close in distance or time zone (or both) [1]. The term 'large' means that a distributed project has a large set of requirements (at least 1000 [31]) with many interdependencies, and is staffed with professionals at least two locations, hence implying more coordination efforts and exposure to risk, because new problems may surface due to the volume of the requirements and the distributedness of the team (and those are easier to handle in small co-located teams). To the best of our knowledge, no research has been focused specifically on understanding how QRs engineering happens in large distributed projects, from the perspective of SAs in near-shore settings.

To this end, we carried out an interview study set out to answer the following research question: *What's the involvement of SAs in engineering QRs for large systems delivered by project organizations in near-shore development centres?*

We report on this exploratory interview-based study with 16 practitioners working both (i) in East-European NDCs of global companies, and (ii) on large software system development projects. The key results of the paper indicate the roles the SAs (in the NDCs) play in engineering QRs, the type of requirements communication processes these SAs are involved in, and the way QRs are discovered, documented, quantified, validated and negotiated. We note that in this study we take the perspective of those SAs that work in the near-shore centers and not the perspective of onshore architects. The results are compared with findings from prior research by other authors and from our own [6,7,8,9,10,26]. We also draw some implications for research and practice.

The remainder of the paper is organized as follows: Section 2 lists related work. Section 3 presents the research plan and its execution. Section 4 presents our results. Section 5 discusses and compares our findings with those in previously published studies. Section 6 evaluates validity threats. Section 7 discusses implications of this research for practitioners and researchers and Section 8 concludes.

2. RELATED WORK

Two streams of related empirical work are relevant to this paper: studies on the software architecture perspective on engineering QRs and on distributed RE. In RE, a 2010 systematic review [13] synthesized evidence from empirical studies on QRs and identified important gaps in our knowledge about how QRs engineering happens in complex environments and how perspectives on this differ (e.g. the authors found no published study on QRs prioritization, and only one study on cost estimation for QRs). No study among those included in this review treated the software architecture perspective on QRs, nor treated QRs in distributed context. Since 2010, progress has been made in the RE community and six studies [6,7,8,9,10,26] focused on the ways in which SAs perceive QRs. The findings of these studies agree that the perspectives of RE specialists and SAs differ regarding the interpretation of the role of QRs in downstream software development activities and the framing and analytical processes important to resolving QRs problems and finding optimal solutions to them. We summarize these studies' findings below.

The survey of Poort at al. [7] with one company's SAs uncovered the relationship between IT project success and the ways in which SAs were dealing with QRs, the importance of QRs as perceived by SAs, and the approaches they used to deal with QRs. This survey found that projects where modifiability is perceived to be of low business criticality "lead to consistently high customer satisfaction" [9]. Also, projects that used QRs verification techniques were more successful than those that did not. In [6], the authors explain the effect of contractual client-vendor relationship on the interaction of software architecture and quantification of QRs. The authors put forward that if information-sharing between parties is limited, this would significantly impede the quantification of QRs. In [26], the authors surveyed the reasoning processes that SAs used while "architecting". The study derived some best practices for architecture decision reasoning aimed at guiding less experienced SAs in three key software architecture activities, namely architecture analysis, synthesis and evaluation. These practices served as the ground for developing a documentation framework for architecture decisions that supports cost-effectively SAs in their projects. QRs were deemed instrumental to it. Next, two other empirical studies [8,9] with SAs in small and medium size projects investigated how SAs dealt with QRs. In these project contexts, the findings revealed the state of the art practices regarding QRs elicitation, modelling, and tool support. The study indicated important gaps between the QRs themes currently researched in the RE community and the state-of-the art industry needs, which motivated the authors' call for more empirical research on QRs from SAs' perspective. Last, a 2013 study [10] explicated the process of engineering QRs from SAs's perspective in large contract-based systems delivery projects. The study found that making SAs an integral part of the QRs processes has been commonplace in contract-based context. The relationship between RE staff/clients and SAs was actively managed whereby the contract meant that SAs embraced responsibilities over QRs. SAs had a well-understood and communicated role to play in QRs engineering and were involved in the discovery, documentation, prioritization, negotiation, quantification, and validation of QRs.

In the area of distributed RE, we consulted four systematic literature reviews on topics related to ours [2,14,15,16]. These were on knowledge sharing platforms in RE [14], on artefact awareness mechanisms [15], on problems and solutions in distributed RE [2], and on distance between RE and downstream activities [16]. We also added the 2014 structured literature survey of Schmid [3], to be sure we cover the most recent publications. The first [14] of these reviews found little empirical data on using knowledge sharing platform in distributed RE. It identified 9 studies on such platforms, only two of which on offshore context (authored by the same researchers). These two studies are on knowledge platform evolution and do not deal in any way with QRs. The second review [15] reports on acquisition and presentation techniques of contextual information when software artefacts are produced in distributed projects. It identified 11 empirical studies focused on documentation as the project artefact for which the process of creating and maintaining awareness among development roles in a project have been investigated in distributed context. Three out of these 11 studies are on RE documentation, however they focus on requirements traceability management and do not go deep into QRs, in particular. The third review [16] indicates that distances to later stages increase the effort needed to align requirements with other development work. The author identified 17 empirical RE studies,

6 out of which are on distributed settings. They deal with tool support, social and technical distances but no study addressed a particular problem or solution in relation to QRs. Last, Schmid [3] reports that empirical studies on distributed RE are inconclusive regarding the added value of specific RE practices, e.g. he found no evidence regarding what requirements elicitation practices add value in what kind of distributed settings. The author explicitly states that the situation in which elicitation problems happen due to the distributed nature of the development, and not because of distributed customer sites, has remained by and large unaddressed. He calls for more primary studies at a more detailed level, so that we can further "in a reliable way the existing state of the art in global RE" (p. 98).

As we found the distributed context to be an under-researched area in both QRs and in distributed RE literature, we felt motivated to initiate research and explore deeper how QRs engineering happens from the perspective of those involved in the field. Findings from such exploration are necessary for advancing the understanding of what RE practices work well in what context.

3. RESEARCH PLAN AND EXECUTION

This study is part of a major initiative to understand how SAs cope with QRs in large and complex projects. The overall objective of our research initiative is to compare and contrast various approaches to dealing with QRs based on specific contextual settings of large projects, e.g. agile, distributed, contract-based system delivery, etc. In the present study, we focus specifically on large projects that take place in a specific kind of organizations and in specific geographic region, which until now evaded empirical RE researchers' attention. More in detail, we consider global companies or major European companies that organize their software development processes by using NDCs located in Easter European countries (e.g. Bulgaria, Romania, Hungary) [11]. Those large companies have evaluated that working with developers in far-off time zones (e.g. China) and in Asian culture might be more challenging than leveraging 'near-shore' opportunities that exist in those countries that only recently became European Union member states. Examples of such companies are Intel, IBM, and the German Telecom [11].

The research objective of the present study is to understand how QRs engineering happens in such a context. Specifically, we aim at answering the following research questions (which also served as an interview guide): **(RQ1)** How do SAs understand their role with respect to QRs engineering in near-shore settings? **(RQ2)** Do SAs and RE staff use different terminology for QRs in near-shore settings? **(RQ3)** How do QRs get elicited in near-shore settings? **(RQ4)** How do QRs get documented in near-shore settings? **(RQ5)** How do QRs get prioritized in near-shore settings? **(RQ6)** How do QRs get quantified in these settings, if at all? **(RQ7)** How do QRs get validated in near-shore settings? **(RQ8)** How do QRs get negotiated in near-shore settings? **(RQ9)** What role does the onshore-near-shore relationship play in the way SAs cope with QRs?

Our research plan was to conduct an exploratory multiple-case study that applies Yin's guidelines [17] and uses structured open-end in-depth interviews with practitioners. We interviewed 16 practitioners (Table 1).

Table 1. Study participants and organizations

ID	System description	# project team members	Project duration in months
P1.	Converged cloud solution	45	10
P2.	Mortgage funds management	21	15
P3.	Online game system	10	18
P4.	Mobile enterprise system	38	12
P5.	Virtualization software	31	10
P6.	Collaborative medical devise management	45	10
P7.	Asset monitoring	40	18
P8.	Global bid preparation system	20	11
P9.	Smart city sensor system	31	12
P10.	Mobile ticket master	23	12
P11.	Financial investment decision support	31	8
P12.	Investment measure tracking	22	14
P13.	Mutual fund management	53	16
P14.	Business-to-business transaction processing for complex product assembling	31	12
P15.	Mobile management of water resources	11	14
P16.	Machine-to-Machine software	18	12

Our research plan included four steps: (1) Compose an interview guide using the guidelines in [18]; (2) Do a pilot interview to check the applicability of the guide to real-life context; (3) Do interviews with practitioners according to the finalized interview script; (4) Sample and follow-up with those participants that possess deeper knowledge or a specific perspective. Each interview lasted 50 to 60 minutes. Each interviewee was provided beforehand with information on the research purpose and the research process.

Choosing the Participants: The 16 SAs came from 10 companies in Bulgaria (4 companies, 9 interviewees), Romania (1 company, 1 interviewee), Poland (3 companies, 4 interviewees), Estonia (2 companies, 2 interviewees). The application domains where the SAs were active included: financial investment, health-care, wireless telecom services, online gaming, converged cloud, and enterprise information systems. The interviews took place between July 15 and Aug 15, 2012. The face-to-face interviews with 7 Bulgarian SAs took place during the first author's visit in Bulgaria, while the other interviews happened over the phone. At the time of the interviews, these SAs were engaged in large [31] projects whose project development teams were formed by near-shore SDC staff and onshore staff located in Western Europe (Germany, Finland, France and Austria). The practitioners were selected because they (i) had professional backgrounds pertaining to our research questions, (ii) were located in the near-shore organizations of their companies, and (iii) had the potential to offer information-rich experiences as they were employed for at least 2 years in their local NDCs. Also, they were genuinely interested in exploring similar questions from their companies' perspectives. All 16 SAs had the following backgrounds:

(1) They all worked in large projects that were running in at least two different development locations, and had clients in more than six countries.

(2) All SAs had at least four years of experience in large systems and at least two years in their local NDCs.

(3) In the projects of all SAs, the work activities were distributed in various ways (Table 2). However the distributions were based on the following commonalities: (i) onshore senior managers were comfortable delegating software architecture design work near-shore, and (ii) near-shore SAs were heavily involved in elaborating QRs and operationalizing them in ways that programmers could understand and do their jobs.

As indicated, the SAs worked on distributed projects which meant that certain parts of the project work (e.g. RE, project estimation) were done in the onshore organization and others (e.g. coding, testing) in the NDCs. While some project roles were clearly assigned to either the NDCs or the onshore organization, other roles were duplicated. Table 2 presents how the projects were organized, which functions were onshore and near-shore and which ones were duplicated. For clarity on what duplication means, we note that a typical breakdown includes a number of near-shored work activities that in some projects have duplicated onshore functions. This means, for the same activity, there are two professionals – one in the near-shore organization and one in the onshore organization, assigned to work in the same role and on the same activity in a project. E.g. Table 2 indicates that in the experience of our participants, a project typically had two professionals on board responsible for project management. Also, two SAs worked on projects where the architecture design function was duplicated (see cell in the last row and the rightmost column), which meant one SA was involved from the NDC staff and one from the onshore IT organization.

We planned the interviews to be 'structured' [18] with regard to the questions being asked during the session. This means, the interviewer (the first author) was the one to control what topics would be discussed and in which order. We note that interview-based exploratory case studies usually are intended to promote self-disclosure and that is what we were after in this work. As in [17], interview studies are not used to provide statistically generalizable results applicable to all people similar to the practitioners in a specific study. The intention of the exploratory case study is not to infer, but to understand. It is not to generalize, but to determine a possible range of views. Therefore, in this study we adopt, based on the recommendations in [17], the criterion of transferability as a useful measure of validity. Transferability asks for whether the results are presented in a way that allows other researchers and practitioners to evaluate if the findings apply to their contexts.

Table 2. Distribution of Work Activities in the projects in which the interviewees worked

	Near-shored functions	Onshore Functions
Typically duplicated (14 out of 16 projects)	- Project Management - Cost estimation - High-level QRs	- Project Management - Cost estimation - High-level QRs
Typical breakdown	- Architecture Design - QRs Operationalization - Coding - Debugging - Stress Testing - Integration Testing - Second line support	- Portfolio Management - Business requirements - User acceptance testing - First line support
Intentionally duplicated on two case study projects		- Architecture Design - Integration Testing

Data Analysis Strategy: Our data analysis was guided by the method for thematic analysis of qualitative data [19] and the constant comparison and coding techniques in Constructive Grounded Theory approach [20].

In essence, this was a process of making analytic sense of the interview data. Constant comparison means that the data from an interview is constantly compared to the data already collected from previously held interviews. After reading each interview transcript, a coding word is attached to a passage of the text – a phrase or a paragraph. The 'code' could be a concept (e.g. 'risk to get out of business'), or an activity (e.g. 'escalating', 'documenting'). Then, in a second step, the codes and the passages coded in that interview were compared to the codes from the previous interviews. We clustered all pieces of text that relate to the same code in order to analyze it in a consistent and systematic way. The results of the data analysis are in Section 4 and the discussion on them and their comparison to previously published findings – in Section 5.

4. RESULTS

Our findings are presented as related to each research question. We supplement the observations with interviewees' quotations. (Below, *P1... P16* stands for the ID of the respective participant from Table 1).

4.1 How do software architects understand their role in near-shore settings?

All SAs thought of their role as *'a senior developer with big picture responsibility'*. This was because they had strong technical backgrounds and spent between 30% and 50% of their work time on programming. When asked about their official job titles, ten of the practitioners had a 'Software Architect' title, while the others had titles as 'Architecture Design Lead', 'Analyst/Architect', 'Senior Programmer/Architect'', 'Tech Lead/Architect'. It's interesting to say that no SA thought of software architecture as a job on which one can be employed 100%.

To get prepared for the work as SAs in their projects, all interviewees have gone through a training program with duration of at least 2 months. Training was organized and delivered by the onshore organization. Its goal was to get near-shore SAs understand processes, standards, principles that were used onshore. The training included (i) principles of architecture design as practiced in the onshore company, (ii) knowledge of the development platform and (iii) knowledge of the application itself (incl. architecture). QRs were explicitly treated as part of (iii), as the SAs were educated on product documentation which they referred to as to "Product Feature Catalogue", "Product Feature Handbook", "Product Quality Handbook", or "Product Quality Manual and Related Features". We note that in a product, a QR is usually implemented in the form of one or more features. Eleven out of the 16 SAs acknowledged that the training was the first occasion for them to develop awareness of the importance of some product-specific QRs (e.g. playability in a game design project). Also the SAs got a chance to network with onshore colleagues and learn from their QRs practices.

How did a SA get assigned to her project? Most of them were put on the projects after completion of the training program and after they have convincingly demonstrated they were knowledgeable enough to suggest options and make decisions in consultation with near-shore or onshore project managers. Our interviewees (the near-shore SAs) indicated that technology platform and

product feature knowledge was considered of higher importance to their work than domain knowledge. Many of them attributed their limited knowledge of the domain to the fact that they had no chance to meet the users of the product to be delivered in the client organizations. This was something that the onshore team members (RE-staff and onshore architects) were involved with: *"We certainly miss the contact with the client. I know no person buys a computer just for the sake of having it, you buy it to use it for something, and non-functionals will be as important to your usage, just as functionality is. But I have no way to guess what's the life of a user using our system… And this is a misfortune". (P6).* Some participants expressed a concern that the lack of domain knowledge hampered them in reasoning about QRs, and no matter what they tried to do to catch up it never worked. For example, a SA working on a mortgage management application project put it this way: *"I'm someone who studied all my life. I earned a PhD in statistics and another in Computer Science, and I know three languages, so it's not a matter not being able to express myself in English, or in German, or in French, however, I have a really hard time understanding mortgage securitization and all the rules that go with it. We have no such industry in our country, our people's life is much simpler, and that's why it's extremely hard to relate to what the onshore friends are talking about. We meet no business users, we even do not know what vocabulary our onshore fellows are using when they are talking with clients on those things. I know nothing about how the client suggests non-functional requirements or what motivates certain levels of security features…And if I do not understand those requirements, how can I explain it to my team here?". (P13).*

As the discrepancies in domain knowledge were perceived a hindrance to both parties, the onshore team missed on a number of opportunities to tap on useful design solutions suggested by local NDCs' SAs: *"I was praised by our onshore manager for being creative and thinking outside the box, but this all came after they screwed up the performance of the system and called us to repair it and do some significant rework. I had been making attempts to talk to them earlier and give them the details on why things would not work as they thought, but no one looked like interested in such a conversation, and I could not talk directly to them and just go over the head of our manager here." (P15).*

4.2 Do SAs and RE staff use different terminology for QRs, in near-shore settings?

Ten out of the 16 SAs in the NDCs had no connection to an onshore RE specialist. They even were unaware of the names of those working as requirements analysts, despite the fact that they knew such roles (e.g. Business Analyst, Business Requirements Lead, Documentation Prime) existed in the onshore organization. These ten SA's were expected to 'get things done as told'. They are supposed to use the terminology being transmitted via the onshore project manager or responsible tech lead. They communicated back to the onshore contacts by using their own near-shore project manager as a mediator. Such a communication process resulted in underestimating grossly the importance of certain QRs as all the SAs were seeing were features, but they failed to understand which feature implemented which QR: *"We receive a list of features and a list of QRs, very high level formulated. But the terms in these formulations are abstract and disconnected to the terms in which the features are defined. Of course, I make my well-educated guesses and use common knowledge of the product and the platform to map the QRs to the operationalizations… those little pieces of functionalities we call features. But at times, I can not go beyond my best guess,… and the result is a best guess too [laughs']. Do not take me wrong, I do not give up, I'm a fast learner, and I'm not shy to ask for advice… but I harbour this feeling of being ineffective in my work, simply because of a broken process of terminology transfer". (P13).*

Three out of these 10 SAs gave specific examples when terminological misunderstandings caused rework, delay, firing a staff member and thus putting the project on hold until a new hire starts. The other 6 SAs knew the RE specialists in their projects and developed gradually the habit to ask for glossary of terms at critical project milestones. Also, they escalated via their project managers to the onshore party.

4.3 How do QRs get elicited in near-shore settings?

We found that no SA was directly involved in QRs elicitation in contact with the client. This was kept strictly with the onshore staff. They however were aware of the iterative nature of the QRs elicitation. Eight out of our 16 participants, were regularly invited to participate in QRs reviews that took place after the initial QRs elicitation conversation happened. These SAs used the Product Feature Handbook as a starting point to understand which QRs could not be mapped against what is known about the product and would call for further elaboration. These SAs then submitted lists of QRs for the onshore RE staff to address in the next QRs elicitation session with the client. This process took different flavours in the different case study projects, e.g. in one case, the project manager mediated the next elicitation gathering session, in another – an onshore SAs was the mediator.

Four SAs indicated that in their project organization there was an institutionalized role of 'Requirements Ambassador' which included a professional responsible for (1) consolidating inputs of RE specialists that are experts in particular sub-domains of the application domain, and (2) communicating to NDCs and clarifying every aspect of the requirements document or any deliverable that comes out of the elicitation. This role had different names in the different projects 'The Req Spokeperson', 'The Messenger', 'The Documentation Prime', but the job duties were very similar and it was the key resource to SAs to participate in the elicitation process as much as their circumstances allowed. *"I've got to ask this guy some days 2-3 times, other days 10-15 times on quality aspects that I could not make much sense of… He knew when I was coming, I had an issue, and over time he learnt what are the kind of questions we are likely to have and tried to address them early on, even before leaving the onshore team to come and spend time with us" (P9).*

4.4 How do QRs get documented in near-shore settings?

All SAs that were interviewed used various standards from the ISO family as the foundation to document QRs. They had templates in various formats approved by their managers for use in their NDCs. These templates were developed at the NDCs themselves without the involvement of the onshore organizations. The motivation for them was the need to have a consistent way of documenting QRs across past and running projects. The onshore organizations were not involved in this. Neither did the onshore organization suggest the NDC use the onshore-developed templates, because these templates assumed certain work practices that were not present in the near-shore organization (for

cultural, monitoring as well as project accounting reasons). The work practices in the onshore organization were 'embedded knowledge' [21] of how work gets done and it was hard to transfer this knowledge to a NDC's professionals. The NDC SAs have been sending the QRs documented in their own formats to their onshore counterparts, who seemed to be happy and appreciative for the documents of the NDC SAs. Also, in the perceptions of our practitioners, it has been cheaper to leave the near-shore project managers and their SAs devise their own templates than to set up and execute an educational program to train near-shore staff on work practices and 'embedded knowledge' that is known to be hard to transfer: "*We have tried to use their templates [of the onshore organization] for any thinkable deliverable, but our fellows [near-shore colleagues] felt it was too voluminous, too many items for things that we never need in our projects. We even could not figure out why they collected (or previewed space for) that much information, if we can use at best 30 % of all that their QRs template covers". (P11).*

While the architects were well aware of ISO standards they did not show any awareness of the difference between product and process standards. We found that the standards that the SAs referred to were belonging to two main streams of standards: (i) management systems as e.g. 9001-27001-20000, and (ii) 'technical standards' (as 14143-x.). The first ones are about 'requirements' to be accomplished and the second ones are about processes. However, SAs used terms from both streams were used interchangeably and it was not clear which QR followed the terminology of which stream. When using the templates, the SAs were stating the QRs in plain natural language text. The templates however helped them ensure abstraction levels are consistent across the definitions of the different QRs (so that the SA avoids being too specific on one QR and too high level on another).

4.5 How do QRs get prioritized in near-shore settings?

All SAs agreed that they make QRs trade-offs as part of their daily job on projects. The prioritization criteria for making the QRs trade-offs were business value, risk, and dependencies between QR-implementation tasks. Business value and risk are evaluated in qualitative terms; whenever possible, the SAs attempted to evaluate QRs quantitatively, e.g. how much it would cost to implement a specific QRs. Business value was judged based on what importance the onshore team places on a QR. If onshore project managers suggest that a product's client values a QR and therefore it's mandatory, 15 out of 16 SAs indicated that they usually do not question the importance. They first attempt to "do it as told", as in the NDC business model there are financial incentives for doing so. Only if 'doing as told' jeopardizes the project, they would approach their project manager who would contact the onshore project manager by means of a formal escalation letter: "*We are trusted that we would deliver what they ask us, that's why we were hired for in the first place. It's a no-brainer… If they say it's important, then it's important and this means end of discussion. Of course if I see QRs that do not go well together, I will let them know and then they will revise their wish list… But basically we are supposed to listen to them and get it done, I mean get the QRs implemented in the resulting product, and not delegating back to them and negotiating. In fact, my tenure here is 8 years and I know of no case when they asked impossible QRs". (P12).*

In the perceptions of those SAs (6 out of 16) who worked in NDCs financially-dependent on the onshore organizations, the premises of the business model were conductive to a work culture where they performed their duties *'as told'*. Their definition of risk meant as Participant P6 put it: "*Risk is the chance that the onshore guys dislike so much what our NDC is doing, that they close it down altogether*." These SAs thought that they internally prioritize QRs based on what their perceived risk is, regarding losing business (or getting out of business altogether) if they do not implement a QR as originally specified by the onshore organization: "*They [the onshore organization] are dependent on us in delivering their promises to their clients. And we are proud of what we do. We are the hands that they use to deliver, and we receive a good reward package for what we do, so why break the collaboration…and why break it just because of debates that presumably will not matter two-three years from now… We can not afford to be negligent about our relationship to the onshore colleagues and loose business because of negligence. I mind my actions and estimate what the impact would be if I leave a QR undone or not done as they wanted it" (P10).* In contrast to this, the ten SAs who worked in NDCs that were profit-and-loss responsible business units, defined risk in terms of project risk (e.g. as in [27]).

Last, we found that all interviewees took a technical perspective on QRs prioritization: they called a priority, what in fact was a dependency between development tasks related to QRs implementation. E.g., if some QRs needed to be implemented and tested prior to others, these were scheduled first in the agendas of both the developers and the testers. Intra-task dependencies were a serious consideration if a QR was found to act on multiple modules developed by different teams, e.g. three SAs pointed out that poor prioritization (aka scheduling of technical tasks) led to delays in user acceptance testing.

4.6 How do QRs get quantified, if at all?

All SAs agreed that expressing QRs quantitatively is a preferable choice. However, they distinguished between two kinds of quantification: for the purpose of demonstrating that the system meets a QRs, and for resource estimation purposes. Quantitative definitions of QRs were provided by the onshore specialists (RE-staff and onshore SAs). In most cases, the near-shore SAs just accepted these definitions and worked with the near-shore project managers to help estimate the effort that it would take to implement them. While no NDC had an official software measurement program in place, the NDC project managers who the SAs worked with had practices for collection and analysis of measurement data. They used a variety of measurement models. This included both product and project metrics. The purpose of QRs quantification was, however, common across all sited where the SAs worked, namely effort estimation. All SAs served as input providers to estimation and participated in reviews of effort estimates pertaining to implementing the QRs in their projects. QRs were quantified in terms that were 'visible' to the onshore management, e.g. number of features that are deemed to implement a QR and must be present in the software system, actual and estimated hours to implement each feature. In other cases, the tasks that the near-shore project manager deemed necessary for implementing QRs were subjected to qualitative estimation in terms of effort, e.g. low effort, medium, high and extremely high effort, which they forwarded to the onshore organization and this input was used for budget adjustment. "*I think this exercise is a necessary part of our NDC's functioning,*

otherwise we have no way to justify why we want the budget that we do. In our business model this is fundamental to our existence as a company in [East European country]". (participant P3).

One SA worked with an Indian development centre of a global company and she identified the IFPUG Function Points to be the standard used in quantifying QRs. However, she did not have any personal experience with this, but provided QRs definitions and operational specifications (descriptions of low level functionalities and architecture design choices) to the project manager in India who engaged a team of three Function Point counters to size the requirements (both functional and QRs).

4.7 How do QRs get validated in near-shore settings?

All SAs were actively involved - by teleconferencing, in formal reviews of the features that would implement a project's QRs. Those six SAs who worked according to the 'do it as told' principle, thought that the onshore contacts (architects, project managers and RE staff) are responsible for validation of QRs. In contrast, the ten SAs who worked and profit-and-loss-responsible NDCs considered themselves responsible of the QRs validation activities. While the first group of SAs thought of the onshore contacts as the QRs owners, these other SAs thought they own what they deliver, features and QRs inclusive: *"If you got to code it, you got to know what you are getting yourself into… and features are important, but these non-functional reqs, I'd say are 100 times more important than features. So, you'd better get them validated, and signed off… Yes, we ask them to do a sign-off…It's much easier to fix a broken feature, but it's a nightmare to fix a non-functional requirement working on many features, and these are critical features…You have no clue where's the bottleneck and you problem-solve heuristically… which is a slow and unproductive process" (P1).*

4.8 How do QRs get negotiated in near-shore settings?

No interviewee seemed to understand the question as negotiation was not a practice in their software delivery models. Nor had they ever heard the term. They thought this was because of their highly technical backgrounds (most SAs were not sure what the day-to-day job of a RE-specialist was like, but they insisted that it's the RE-staff who should know about negotiation). Ten SAs thought of negotiation as part of QRs validation: *"I provide information for them to make better decisions, and they use my input to what you may call negotiation with other onshore stakeholders. I think my part is an informant, and that's what they thank me for." (P2).*

Three other SAs indicated that there were incentives at play in their organizations that favour the situation that they remain silent and let the onshore team discover that existing architecture has no way to meet critical QRs, e.g. in projects where an aged system was brought to them to revamp and maintain it. *"You do not want to be the bad messenger; you've got to do the work and get what they give you, and let them figure it out for themselves. I will tell them if they ask me, but I would not take the liberty to write to someone onshore, because I know I'm not supposed to do so. Let them make their own decisions… we are only doers here". (P11)*

4.9 What role does the onshore/near-shore relationship play in the SAs coping with QRs?

Were the SAs thinking that their handle of QRs was attributable to the fact that they were on the near-shore side in a global company? All 16 SAs perceived this was the case. They thought of the business model behind a NDC in a country as the most important aspect that "*silently infused"* their behaviour. In the words of participants P1, P4 and P12, respectively: *"You get paid to get the job done. Their job done. You feel it in the air". (P1) "If you you're your next paycheque does not depend on their mercy (I'm kidding), then you can act differently. For example, I think I will play an active role in QRs gathering, I'd love to meet the clients, and if my organization would be able to pay for my visit to the clients' sites and learn how the clients use what we are coding, then of course I will have a different outlook to life at our NDC…" (P4). "I have trouble understanding the mortgage securitization stuff. If I were in their country, I would learn it and use this knowledge more effectively in my work. I'm now guessing mostly and this would'nt happen if I had their full knowledge of the domain." (P12).*

All SAs agreed that the relationship played an important role. All also agreed that some activities were more affected than others. These SAs thought that the professional behaviour was a result more of dominant business culture at the near-shore workplace (e.g. clique-building) and strong tendency to corrupted mentality (e.g. deliberately assigning convenient people to positions regardless the relevance of their background and fit to the job), than the business model itself. *"In such a workplace, by default, you are not supposed to negotiate QRs, or even raise your voice if you see a requirement is not OK. You have a very strong case to initiate any escalation to your manager." (P16).*

Our interviewees mentioned that onshore RE-staff acquires domain knowledge not only as a result of training but multi-year interaction with users. Although the near-shore SAs had education from their countries most prestigious universities, their background was technical and they felt a need to complement it with knowledge of the domain. While they felt well about making technical decisions without knowing the domain, they mentioned cases when their work was impeded as a result of missing domain knowledge. In two occasions this resulted in major system redesign, in another – in a complete abandoning of the developed component and in fully re-coding it from scratch.

5. DISCUSSION

This section compares and contrasts our finding to previously published ones. It is organized by the themes in our RQs.

SAs' understanding of their role. All SAs thought of their role in technical terms, e.g. 'a senior developer with big picture responsibility'. Unlike in [10], where architects reason about themselves as communicators across multiple stakeholders including clients and users, our interviewees understood themselves as 'doers' or 'technical heroes' who make their best to achieve the result wanted by the onshore team. We found a prevailing 'do as told' work attitude. This was especially important to those SAs who worked in NDCs where the onshore organization fully controlled the financial resources. This finding agrees with empirical results by Levina et al. [22] who researched offshore collaborations with India and Russia.

Furthermore, our results suggest that NDCs are tasked with complex projects that are hard to codify. However, instead of

attempting to transfer as much domain knowledge as possible, in the perceptions of our interviewees the onshore organization focused on leveraging the near-shore SAs' technical skills as much as possible. One explanation for this choice, according to some SAs, was the business case for near-shoring: acquiring domain knowledge is a time-extensive process and knowledge transfer therefore might be more expensive than what the business case for near-shoring implies. We think that clarifying whether this is indeed so and under what circumstances, forms an interesting line for future research.

Do SA's use different terminology for QRs? Our results suggest this is a serious issue. This finding agrees with [9] where SAs collectively indicated a broad terminological gap between SAs and RE-staff. We assume that this agreement may well be due to the fact that the interviewees in both studies had strong technical background and had no full-time jobs as SAs per se, but performed other functions, e.g. programmers. Also, in our study, the gap in terminology is traceable back to the fact that the business domain knowledge was the aspect in which in NDCs and onshore IT organizations differed most.

QRs Elicitation. Our study revealed SAs only passively participated in elicitation. This contrasts [10,26] that found SAs were actively involved. Our SA's participation was mediated by the Requirements Ambassador role designed especially for this purpose. It is a resource to the NDC's SAs to escalate, clarify and resolve any issue pertaining to QRs.

QRs Documentation. We found that templates plus natural language were used most in NDCs. However, the onshore company's templates were perceived as only marginally useful, as SAs did not have to deal with much of the information included. Simplified templates were created at the initiative of some SAs and based on ISO standards. While this agrees with [10], it contrasts [9] where the SAs could not agree on one specific systematic way to document QRs and natural language was the only common practice used. Why this difference occurs? We assume that it's because of the fact that the NDCs needed (1) to demonstrate compliance to what the onshore organization expected them to do and (2) to make it transparent for the onshore organization to track what the NDCs did and how. We think it's realistic to expect that in such contexts, development progress is monitored on an on-going basis by means of explicitly defined indicators, and this forces IT professionals to adopt a sound template-based documentation flow throughout the project [21].

QRs Prioritization and Negotiation. A key finding is that prioritization of QRs is understood in narrow and strictly project management and technical terms, namely as task dependencies and, in turn, as 'work scheduling'. The SAs' thinking of priorities could be traced back to the dominating mentality in NDCs to get the job done 'as told'. We also found that negotiation was by and large an unknown concept. This contrasts previously published observations [10] that SAs actively stepped in to negotiate QRs.

QRs Quantification. In contrast to [10] where SAs were involved in quantification for the purpose of QRs verification, this study found that QRs quantification in near-shore settings served estimation purposes. Expert judgements were considered inputs to the budget allocation process. The NDCs however did not have any fixed guidelines on how to execute an expert-judgement-based estimation process. They seemed aware of its importance and were committed to it. One could think that this is because of NDCs needs to justify their budget and to carefully estimate any piece of work so that they do not run on empty.

QRs Validation. We found that the attitudes and the level of participation of SAs in validation differed based on whether or not they had any control over the outcomes of the requirements validation process. The SAs who were paired with a SA in the onshore organization were the most actively involved in validation. Also those SAs who worked for profit-and-lost units deemed their participation 'active'. This makes us think that the SAs' motivation and interest in validation grows with the increasing level of control over their work. We consider this a working hypothesis worthwhile investigating in future studies.

Role of the onshore/near-shore relationship the way SAs cope with QRs. Our study supports the notion that the onshore/near-shore relationship orchestrates the work processes of SAs while coping with QRs. We found that there were at least three dimensions based on which the SAs reasoned about how they executed their QRs tasks: (1) the business model underlying the collaboration between NDCs and headquarters; (2) the presence or absence of a mediator (e.g. a paired SA or a Requirements Ambassador) from the onshore organization, and (3) culture and mentality in a particular location. We however could not claim these three dimensions are orthogonal. Instead, we think they overlap. However, we think that only through follow-up studies on this topic could we collect more evidence that clarifies the exact relationships between the dimensions.

We noticed that the level of social engagement of the SAs in the QRs tasks varied widely. This being said, we found that one could distinguish between levels of individual SA's engagement. Drawing on Barki's and Hartwick's [23] user involvement framework, we distinguished two levels: (1) participation in an ac-activity, which is evaluated by the extent to which an individual performs a specific QRs-related task, and (2) involvement in an activity, which requires a mental state of identification with some goal or artefact, to an extent that it is perceived as both important and personally relevant. Table 3 presents the QRs activities and whether the evidence hints to participation or to involvement. We make the note that QRs negotiation tasks are not included in Table 3, as no SAs signalled awareness of what these tasks are. Also, we note that regarding QRs prioritization, the SAs are both 'participating' and 'involved': they participate in defining dependencies among development tasks needed to fulfil the QRs, however they are involved when it comes to identify, and make decisions of trade-offs among QRs. The mapping in Table 3 has some implications for research and practice. Existing research [6,7,8,9,10,26] casts the impression about SAs are active agents in RE. Our study shows that such a claim (that SAs are active) assumes the presence of some contextual factors. What these factors are could become known only by further research. Our findings give a first indication only that the extent to which the underlying business model, the local culture and the relationship

Table 3. Mapping of RE tasks against participation and involvement

RE tasks	Participation	Involvement
Elicitation	+	
Documentation		+
Prioritization	+ (task dependencies)	+ (QRs trade-offs)
Quantification	+	
Validation		+

to the onshore organization influence the participation and involvement of SAs in near-shore development projects. We however think that there might be more factors to be discovered (and also these three could be refined) to get a more complete understanding of why SAs in NDC do what they do in engineering QRs, and in turn, how to leverage existing RE tools to better support what they do.

The high level of involvement in documentation, QRs trade-offs, and QRs validation tasks indicates that the work of SAs in engineering QRs in NDCs, is mostly centred around those parts of the RE process that in essence make sure that what the SAs have to deliver and commit to is well documented and well understood. We think this is because documentation and validation are of paramount importance for the downstream development activities for which the local project managers and software development teams are directly responsible. One could consider this as a way of the NDC's SAs to manage project risk. As a 2012 empirical study on the relationship between user risk and requirements risk on process performance in IT project [24] concludes: "managers should do their best to insure that core requirements are carefully identified at the outset of the project" (p.15). If the NDC has the incentive to bring out the best possible work (because this is crucial to the organization's financial health), then it would not be surprising that well-understood QRs are key for them to deliver on this promise. We, however, observed that the documentation approaches were created and used at the initiative of the local NDCs. This means, the onshore organization might have been unaware of their extensive documentation needs or might have found it overly expensive to adapt and introduce their documentation process in a particular location in a particular country. We consider this point to be interesting for future research, so that we learn more about the possible alignment scenarios between QRs documentation processes and those at the level of NDC.

6. VALIDITY THREATS

Our evaluation of the possible threats to validity of the observations and conclusions in this study used the checklist in [25]. As we executed exploratory qualitative research, the key question to address when evaluating the validity of its results, is [25]: to what extent can the practitioners' experiences in coping with QRs be considered representative for a broader range of projects, companies, application domains? Of course, we can not claim that our case study projects would be representative for all the possible ways in which QRs engineering happened in NDCs. Following [5], we think that it could be possible to observe similar experiences in projects and companies which have contexts similar to those in our study, e.g. where (i) NDCs share the same business model, i.e. the near-shore organization is fully dependent on the onshore one in financial terms, (ii) the NDCs are in the same Eastern European countries and share the same organizational culture, and (iii) the SAs have absolutely no exposure to clients and take a technical view on what they do. As suggested in [5], "if the forces within an organization that drove observed behaviour are likely to exist in other organizations, it is likely that those other organizations, too, will exhibit similar behaviour" (p.12).

We also accounted for the inherent weaknesses of interview techniques [18]. A threat is the extent to which the SAs answered our question truthfully. We took two steps to minimize this threat by (i) recruiting volunteers under the assumption that if a practitioner would not be able to be honest, he/she could decline his/her participation at any stage of the research and (ii) that we ensured no identity-revealing data will be used in the study. Next, a well-known threat in interview studies is that an interviewee has not understood a question. However, we think that in our study, this threat was not significant because: (i) in 9 cases, the researcher has been known the SAs for more than 20 years and they felt at ease to ask for clarifications. Also, the researcher talked to these 9 SAs by using their native language, and the SAs shared more than they would have done if they would have talked to a stranger in English; (ii) in the other cases, the interviewer asked about the same topic in a number of different ways. An example of this was the question about requirements negotiation, which required more explanation from the researcher as it was clear early in the conversation that the SAs were unaware of the meaning of this term. Next, we accounted for the possibility that the first author might have instilled her bias in the data collection process. We followed Yin's recommendations [17] in this respect, by establishing a chain of findings: (i) we included participants with diverse backgrounds (i.e. industry sector, type of system being delivered), and this allowed the same phenomenon to be evaluated from diverse perspectives (data triangulation [18]); and (ii) we had the draft study report reviewed by the SAs.

7. CONCLUSIONS

We found that engineering of QRs in near-shore settings is a recurring process where the quality of the output at the NDC is contingent on the quality of the input provided by the onshore organization. Measurable QRs for which resources, time in the schedule and budget are allocated have to be monitored and updated on regular basis as more information gets available in the project. The synchronization between the most recent available information in a project and the values set for QRs plays a fundamental role for the successful delivery of these requirements in distributed projects. QRs expressed in terms of values to achieve are of highest benefit for the project team in the very first project stages, when most critical decision-making takes place, such as the decisions on budget allocation. This is especially important in NDCs that financially depend on the onshore organization or the headquarters. We note that this brings out an interesting observation: the inputs into QRs definitions (i.e. something that happens at the onshore organization) are getting better only when he outputs are starting getting worse (i.e. when the system being developed at the near-shore organization exhibits suboptimal values on key QRs). This looks like a dilemma for project managers because they may want to organize for a more effectively executed communication processes. Addressing this dilemma seems to be of paramount importance to finding out how to resolve the problem of dissonance in understanding of QR between NDC's SAs and onshore staff, and in turn the problem of poor estimation in the early project stages of the resources to be allocated to the implementation of QRs.

Our findings have the following implications: To onshore SAs/ RE-staff, it suggests that there might be several risks related to QRs. These seem to be typical for the near-shore settings, as we did not observe them in our previous interview study [10] in large-scale contract-based projects in Western Europe. Such risk factors we found are: (1) The lack of possibilities for joint QRs prioritization and negotiation, between RE-staff and SAs. Due to the financial dependency of the near-shore partner, they are not supposed to discuss QRs and do not dare to do so. Like this, their perspective and experience is not taken into account during decision-making about QRs which can lead to suboptimal RE decisions. (2) The lack of domain knowledge on the side of the

near-shore SAs, terminology gap and the lack of contact to customers can lead to inappropriate design decisions. (3) The NDCs' SAs do their best to fulfil the QRs which have been defined without them, and to document them in a consistent form appropriate for them. However, the collaboration could be improved by integrating SAs better in the decision-making processes in requirements elicitation/prioritization/ negotiation. To RE researchers, our study suggests that to understand how variations in context result in variations in the ways in which SAs approach QRs, it seems worthwhile considering possibly borrowing theories from other disciplines (e.g. behaviour science) that help explain how context influences professional behaviour. E.g. how an existing business model for near-shore development shapes the behaviour of RE staff and SAs is an interesting question demanding future research.

In our immediate future, we plan to use our results in follow-up studies, in other countries known for concentrations of NDCs, e.g. Turkey. Comparing our results with findings in zones outside Europe would benefit both onshore and near-shore businesses in companies on the extent to which handling QRs might depend on country-specific types of business models. Also, it would be interesting to understand how our results would compare to findings from companies involved in offshore insourcing.

8. REFERENCES

[1] Carmel, E. and Abbott, P., 2007. Why 'nearshore' means that distance matters. *Commun. ACM* 50(10): 40-46.

[2] Ebling, T., Audy, L., Prikladnicki, R. 2009. A systematic literature review of requirements engineering in distributed software development. In *Proc. ICEIS* (Milano).

[3] Schmid, K. 2014. Challenges and solutions in global requirements engineering – A literature survey. In *Proc. SWQD* (Vienna, Austria), 85-99.

[4] Bass, L., Clements, P., and Kazman, R. 2003. *Software Architecture in Practice.* Addison-Wesley, USA.

[5] Seddon P. and Scheepers, P. 2011. Towards the improved treatment of generalization of knowledge claims in IS research: Drawing general conclusions from samples. *European Journal on IS*, 21, 6-21.

[6] Poort, E.R., Key, A, de With, P.H.N., and van Vliet, H. 2012. Issues dealing with non-functional requirements across the contractual divide. In *Proc. WICSA/ECSA* (Helsinki).

[7] Poort, E.R., Martens, N., van de Weerd, I., and van Vliet, H. 2012. How architects see non-functional requirements: Beware of modifiability. In Proc. REFSQ (Essen, Germany).

[8] Ameller, D. and Franch, D. 2010. How do software architects consider non-functional Requirements: A survey. In *Proc. REFSQ* (Essen, Germany), 276-277.

[9] Ameller, D., Ayala, C., Cabot, J., and Franch, X. 2012. How do software architects consider non-functional requirements: An exploratory study, In *Proc. RE Conf.* (Chicago, USA).

[10] Daneva, M., Buglione, L., and Herrmann, A. 2013. Software architects' experiences of quality requirements: What we know and what we do not know? In *Proc. REFSQ* (Essen).

[11] ECONOMIST (2005) The rise of nearshoring. *The Economist* 377(8455), 65–67.

[12] Verner, J. and Evanco, W. 2005. In-house software development: What project management practices lead to success? *IEEE Software* 22, 1, 86-93.

[13] Bentsson-Svensson, R., Höst, M., Regnell, B. 2010. Managing quality requirements: A systematic review. In *Proc. EUROMICRO-SEAA* (Lille, France), 261-268.

[14] Sillaber, C. and Breu, R. 2014. The impact of knowledge sharing platforms in distributed requirements engineering scenarios: A systematic review. In *Proc. KMO* (Kaohsiung).

[15] Vivian, R., Huzita, E., Leal, G., and Steinmacher, A. 2011. Context-awareness on software artefacts in distributed software development: A systematic review. In *Proc. CRISG* (Paraty, Brazil), 30-44.

[16] Bjarnason, E. 2013. Distances between requirements engineering and later software development activities: A systematic map. In *Proc. REFSQ* (Essen, Germany).

[17] Yin, R.K. 2008. *Case Study Research: Design and Methods*. Sage, Thousands Oaks, USA.

[18] King, N. and Horrock, C. 2010. *Interviews in Qualitative Research*. Sage, Thousands Oaks, USA.

[19] Saldana, J. 2010. *The Coding Manual for Qualitative Researchers*. Sage, Thousand Oaks, USA.

[20] Charmaz, K. 2007. *Constructing Grounded Theory*. Sage, Thousand Oaks, USA.

[21] Nicholson, B. and Sahay, S. 2004. Embedded knowledge and offshore software development. *Information and Organization* 14, 4, 329-365.

[22] Levina, N. and Vaast E., 2006. Innovating or Doing as Told? Status Differences and Overlapping Boundaries in Offshore Collaboration. *MIS Quarterly* 32(2): 307-332.

[23] Barki, H. and Hartwick, J. 1994. Measuring user participation, user involvement, and user attitude. *MIS Quarterly* 18, 1, 59-82.

[24] Keil, M., Rai, A., and Liu, S. 2012. How user risk and requirements risk moderate the effects of formal and informal control on the performance in IT projects. *European Journal on IS* 22, 650-672.

[25] Runeson P. and Höst, M. 2009. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering* 14, 2, 131-164.

[26] Heesch, van, U. and Avgeriou, P. 2011. Mature Architecting - A Survey about the Reasoning Process of Professional Architects, In *Proc. WICSA*(Boulder, CO). 260-269.

[27] Ropponen, J. and Lyytinen, K., 2000. Components of Software Development Risk: How to Address Them? A Project Manager Survey. *IEEE Trans. SE*. 26(2): 98-112.

[28] British Standard BS ISO/IEC 25030:2007: Software engineering — Software Product Quality Requirements and Evaluation (SQuaRE) — Quality requirements.

[29] IEEE: SWEBOK V3.0 Guide to the Software Engineering Body of Knowledge, 2014, www.swebok.org

[30] http://v-modell.iabg.de/v-modell-xt-html-english/

[31] Regnell, B., Berntsson-Svensson, R., and Wnuk, K. 2008. Can we beat the complexity of very large-scale requirements engineering?. In *Proc. REFSQ* (Essen, Germany).