'Negotiating Knowledge': The Case of a Russian–Norwegian Software Outsourcing Project

Vegar Imsland
University of Oslo, vegari@ifi.uio.no

Sundeep Sahay
University of Oslo, sundeeps@ifi.uio.no

'Negotiating Knowledge': The Case of a Russian–Norwegian Software Outsourcing Project

Vegar Imsland & Sundeep Sahay

Department of Informatics, University of Oslo, Norway
vegari@ifi.uio.no & sundeeps@ifi.uio.no

Abstract. This paper presents an empirical analysis of a global software development relationship between a Norwegian client and Russian contractor for the redesign of a payroll system called SalarySystem. The empirical analysis, which involved multiple visits to both the Russian and Norwegian sites and meetings and interviews with people from different levels involved with the system, revealed some interesting insights into how the project was initiated, how it nearly degenerated into a breakdown situation, and how learning took place and the project was first salvaged and then was grown. The theoretical notion of embedded knowledge (Nicholson and Sahay 2004) is drawn upon, and extended to describe features of embedding arising from *domain*, *language*, and *project management* issues. Four mechanisms through which this embedding was negotiated are discussed: use of TestTool; use of test cases; use of ICQ; and increased face to face interaction.

Key words: global software outsourcing, knowledge, embedded, Norway, Russia.

1 Introduction

Global Software Outsourcing (GSO) relationships refer to the conduct of software development work by firms or groups spanning national boundaries and supported by the use of ICTs. The practice of GSOs has been largely normalized in North America where firms are routinely getting their software requirements being met through firms based in low cost countries like India and Russia (Carmel 1999). However, European countries, especially those in Scandinavia, have typically been much slower in engaging with GSO work for various reasons including more protective labor laws, diversity of languages used (other than English), smaller scale of projects, and generally seen to be more inward looking in their approach to globalization (Imsland et al. 2003). However, in recent years, even the Scandinavian countries are facing pressures of not enough numbers of technical graduates and rising costs of operations, making them seriously consider global options for meeting their software development needs. For example, the Norwegian Prime Minister led a delegation of 30 ICT firms to India in order to identify software business relationships, and in February 2005, the Norwegian Minister of Trade and Industry made a similar trip. In 2004, the Norwegian ambassador to India expressed concern that while countries like USA, England, Germany, and France were gaining experience and contacts within this software domain, Norway was dragging behind. The Ambassador also noted that the Norwegian firms doing outsourcing were focusing primarily on simple maintenance contracts typically for single projects, and still not seeing the benefits of investing in long term relationships (Verdens Gang 2004). A survey carried in Mandag Morgen (2004) concluded that 80.000 Norwegian jobs (in all sectors) have been moved abroad the last five years, and that 80.000 more jobs are expected to be moved during the next five years. The main reasons cited for moving the jobs were lower costs, access to new markets, and gaining access to better competence.

The significance of the GSO phenomenon can be gauged by the fact that it has even become an important debate item in the elections in both the US and UK. In the run up to the US elections, while Kerry threatened to end tax breaks for firms outsourcing jobs, Bush, on the contrary, promised greater tax breaks. American unions were critical of Bush's policy of encouraging outsourcing when the national situation was characterized by unemployment figures of nearly 10 million. Mr. Kiran Karnik, President, NASSCOM while pointing out that India has taken more column inches in the US presidential elections (referring to the media reports on outsourcing), described the election debate on outsourcing as *mostly rhetoric*. Consulting firms, for example

Forrester Research, Inc., has estimated that the demand for offshore outsourcing will account for 28% of IT budgets in Europe and the U.S. in 2004, and offshore IT workers worldwide will grow from 360,000 in 2002 to more than 1 million by 2005 (http://www.intsoft.spb.ru). Large software firms in India like InfoSys and IBM plan to hire about 10 000 developers in 2004. This continued business interest is reflected in the fact that many countries, for example, Vietnam, Philippines, Iran and Ukraine, are taking proactive steps to establish policy and infrastructure to attract global customers (Carmel 2003).

In this paper, we focus on a Norwegian company engaged in a GSO relationship with a Russian partner. The focus is important for two reasons. One, the practice of outsourcing is far from normalized in the industry here, even though there are sporadic examples of Norwegian firms doing such work (for example, large ones like DNV and small ones like OrgConsult). During a workshop conducted in Oslo in 2001, we heard presentations from many Norwegian firms who expressed their interest in adopting such arrangements but felt lacking in know how and experience of engaging in an activity perceived as being rather risky and requiring larger scale projects and investments than the level of their current operations. These firms were keen to learn from experiences of other Norwegian companies on the nature of practical challenges and approaches to make GSOs work in practice. Two, while there are many studies of North American firms doing outsourcing (for example, see Sahay et al. 2003), there are almost no studies of Norwegian or Scandinavian firms. Empirical studies of GSOs have established the extreme complexity of managing these relationships due to reasons of cultural, geographical and temporal separation. Since a Norwegian-Russian relationship provides a rather unique cultural and geographical context, insights generated from such an analysis can contribute to a broader understanding of the complexities of GSOs, especially related to the management of cultural diversity.

The empirical basis for our analysis emerged as a result of the Oslo workshop referred to earlier where a call was made for more empirical studies of Norwegian firms. This led to the initiation of two case studies, the story of one of which is reported in this paper. Broadly, the aim of this paper is to empirically examine the challenges related to managing knowledge GSO relationships involving Norwegian firms, especially as seen from the perspective of managers responsible for the Norwegian operations. The challenge of managing knowledge has been highlighted by researchers (for example, Nicholson and Sahay 2004) as being fundamental in GSO relationships. These authors develop the concept of *embedded knowledge* to argue that knowledge is not fully mobile as is commonly assumed, and have argued about the context specificity of knowledge and the ongoing processes of negotiation between the

V. Imsland & S. Sahay • 103

involved partners through which knowledge is understood, shared and used. This paper seeks to analyze these contextualized processes around managing knowledge in a Norwegian-Russian relationship. More specifically, the two research questions addressed are:

- What are the complexities around the management of knowledge in the context of a Norwegian-Russian GSO relationship?
- How can these processes be managed more effectively?

The rest of the paper is organized as follows. In the next section, we discuss the challenges of managing knowledge and its significance in the context of GSOs. This is then followed by the description of the empirical approach adopted, and in section 4 the case study is described. In section 5, the inductively developed theoretical themes are described, followed by some brief conclusions in section 6.

2 The Complexity of Managing Knowledge in the Context of GSOs

Software development represents a complex problem solving process simultaneously involving multiple individuals, teams, and organizations with competing goals, interests, and responsibilities (Curtis et al. 1988). Software development is a knowledge intensive activity in which designers, developers, project managers, and end users each influence the process in different ways, bringing into play a multiplicity of knowledge systems, and ways of understanding and communicating them. This multiplicity is magnified in distributed settings (as involved in GSOs) as compared to software development taking place in co-located settings where the actors are relatively proximate in geographical terms. In co-located settings, there can also be expected to be greater cultural proximity as compared to in GSOs which by definition involve the coming together of actors from different firms and countries. This cultural diversity contributes to significant management complexities relating to managing knowledge.

Given the people intensive nature of software development, a key challenge concerns how knowledge is shared across the different groups of people. While a common assumption made in the conduct of GSO work is that knowledge can be freely transferred across geographical, physical and cultural boundaries, Nicholson and Sahay (2004), based on an empirical analysis of a UK-Indian software development project, develop the concept of embedded knowledge to emphasize also its non-migratory features, and the organiza-

tional processes and practices that contribute to this. By being embedded in cognitive, organizational and societal conditions, knowledge often tends to be non-migratory (Bechky 2003; Lam 1997; Lam 2000; Szulanski 1996; von Hippel 2001). Software development, especially in GSO settings, brings into play both migratory knowledge residing in *mobile packages* such as books, formulas and machines, and embedded, non-migratory knowledge that "resides primarily in specialized relationships among individuals and groups and in the particular norms, attitudes, information flows and ways of making decisions that shape their dealings with each other" (Badaracco 1991, p. 79).

Embeddedness creates difficulties in developing mutual knowledge and contributes to significant management challenges, for example related to conducting design and requirements analysis (Cramton 2001; Robey et al. 2000). Nicholson (1999) provides an interesting example of the problems in understanding requirements in a project involving an Indian firm developing an application for UK social security benefit. In India, where the concept of a State supported welfare system does not exist, the developers struggled to understand how such a welfare system worked in practice, for example, who the users were and how they interacted with the system. This knowledge, deeply embedded in existing UK institutions, was assumed by the UK group to be mobile and which could be easily understood by the Indians based on formal documentation. In the absence of such mutual knowledge, and understanding about what the other side knew or not, there were problems in conducting requirements analysis, and contributed to delays and frustrations on both sides.

While on the one hand, it is important for developers to have an integrated knowledge of the whole system to be effective, on the other hand, GSOs apply a contrary approach in which a piece of work is split into smaller modules, distributed for development across multiple sites, and subsequently integrated. These processes of splitting, distributing and integrating bring into play different kinds of knowledge at varying stages of the project. At the initial stage of design, knowledge about methodologies for design and requirement analysis are required, but as the project moves into the coding phase software project management techniques need to be well understood. After the completion of the development, knowledge about testing and integration becomes crucial. Knowledge demands also vary with types of projects, for example structured work like maintenance and bug fixing requires primarily technical knowledge about tools and platforms, while design work places additional demands to gain knowledge about management practices and user needs.

The levels of knowledge and the costs of attaining them need to be judiciously balanced between *onsite* (when the developer is at customer site) and

V. Imsland & S. Sahay • 105

offshore (developers in their home country) (Sahay et al. 2003). Since GSOs involve the splitting of work between onsite and offshore, coordination of development requires mutual knowledge between the two sides so as to allow for effective transfer of tasks, instructions, and finished pieces of code. To maintain this knowledge balance which is dynamic and having different cost implications (an onsite developer typically costs more than one offshore), firms have developed various practical strategies. For example, by keeping more people onsite when the knowledge levels are low, and slowly reducing the onsite presence and moving a higher proportion of the work offshore. There are also key *bridge people* who are required to straddle both cultures (for example, by being bilingual) and serve as the conduit for communicating and coordinating knowledge (Krishna et al. 2004).

Knowledge sharing is fundamentally dependent on communication, with and without ICTs. Building shared understandings, for example, between people from *high-* and *low context cultures* (Hall 1984) is challenging because of varying communication styles. In low context cultures, communication tends to focus on the spoken word and the success of the interaction rests primarily on how clearly the sender of the message is articulated. In contrast, high context communication is implicit in nature (Nance and Strohmaier 1994), including taken-for-granted information that have to be interpreted in conjunction with the spoken word. While these challenges are present even in face-to-face interaction, they are further magnified in GSO work because of the heightened cultural diversity of those interacting and also due to the mediation of ICTs. The use of technologies like the internet is socially and culturally defined, contributing to varying patterns of use in different settings (Castells 2001). For example, in the context of GSOs, Krishna and Sahay (2001) describe differences between how Japanese and Indians used the e-mail differently which influenced their joint development project. While Indians tended to be rapid in writing e-mails and expecting replies, the Japanese due to their emphasis on detail and completeness, and their relative weakness with the written English language, tended to be much slower to respond to a message. While the Indians interpreted this time lag to imply that the Japanese were casual or did not understand software development, the Japanese felt that the Indians were being too pushy in their interactions and expectations.

The mediation of ICTs in GSO work also contributes to different challenges arising from the complexities of technology, diversity of standards, and varying socio-cultural contexts. Some authors (for example, Carmel 1999) argue that the power of ICTs enables software development to take place in dispersed teams, and projects can *follow the sun*. An opposing view is that distributed development is inherently ineffective since "project costs increase in

proportion to the time it takes for people to understand each other," (Cockburn 2002, p. 81). The time taken for effective understanding to develop is related to the proximity of the people involved notwithstanding the use of ICTs. A middle view is presented by Maznevki and Chudoba (2000) who argue that in the initial stages it is important to have face-to-face interaction which can allow the development of a temporal rhythm that can help support subsequent ICT-mediated communication.

Carmel (1999) argues that a tactical approach to reduce knowledge complexities is to outsource projects that reduce the dependence on the communication of *local knowledge*, and by reducing the cultural distance through choosing an outsourcing partner from a country culturally similar to ones own. For example, some Norwegian firms may prefer to work with Russians because of the perceived similarities of the European mindset rather than with the Indians who are seen to be more different. One of the most cited reasons for the Indian success has been English language proficiency of their developer work force (Arora et al. 2001) and their relative ease in working with North American and British clients (as compared with China). Despite the availability of a number of computer supported communication tools, reducing temporal distance (related to time zones) of the outsourcing partners is also seen as a simple and effective strategy to deal with challenges of knowledge sharing. Certain forms of tasks (say related to developing design specifications) may be best performed in synchronous (telephone, videoconferences, chat-sessions and of course face-to-face) modes of communication because of the increased need for local and context specific knowledge, while asynchronous communication (like e-mail) is more useful to transmit reports or perform logistical and relatively context independent tasks like coordinating a videoconference meeting (Sarker and Sahay 2004). The advantage of reducing time-zone differences as a strategy to manage knowledge is reflected in the growth of what is called *near shore* operations by firms in the Caribbean or Mexico selling their advantage of being in the same time zone as the United States (Abbot 2004).

We have highlighted some of the challenges associated with managing knowledge in GSO, especially trying to emphasize the context-specificity of knowledge and the processes through which they are constructed, shared, and interpreted. A Norwegian-Russian context thus provides a unique setting, and understanding the knowledge related challenges is a key focus of this paper. We next described the methodology followed, and then the case description.

V. Imsland & S. Sahay • 107

3   Methodology

The research is based on the study of a GSO between ScanSys (the Norwegian vendor) and RussCo (the Russian contractor). ScanSys is a company with 13 000 employees located across Europe, headquartered in Finland, and with a primary Nordic focus. ScanSys' main business is in providing consultancy services on business systems to its customers. RussCo, with 110 employees and situated in St. Petersburg, provide offshore software development services primarily to American clients, but now interested in establishing a stronger client base in Scandinavia. RussCo, together with approximately 20 other software companies in St. Petersburg, form the Fort Ross Consortium that aims to promote member companies through developing marketing contacts and lobbying with the government. The first contact between ScanSys and RussCo was enabled through this consortium in 1999, and after a preliminary analysis by ScanSys of potential partners, a GSO was established in 2001. One of the first projects given to RussCo was called SalarySystem, a payroll management system used by nearly 11000 small and medium sized companies in Norway. The project involved reengineering SalarySystem from its existing expensive, licensed, and soon getting obsolete database, to the new, free of charge and more powerful MySQL platform.

Our research focusing on SalarySystem started in August 2002, six months after the initiation of the SalarySystem project, and lasted for 12 months. Preliminary discussions at ScanSys indicated that this project had experienced many problems since its launch in 2001, leading to delays, frustrations, shift of personnel, and more positively, also to significant learning for both sides about how to handle future projects. Our research aim was to understand the nature of the problems, why they emerged, how they were dealt with, and what learning emerged for both sides from this experience. As the research unfolded, the challenges relating to managing knowledge became evident, and our research questions also became increasingly focussed on related issues.

A mix of data collection strategies were used including interviews, participation in project meetings, study of documents (such as the project plan), and systems demonstrations. We prepared management reports, and feedback obtained by respondents from both sides also provided useful insights into the perceived coherence of our interpretation. During the 12 months of the research, we visited the ScanSys office in Oslo several times, and the support office and subsidiary at Gran and Sandnes respectively once each. The RussCo office in St. Petersburg was visited twice. In addition to the formal interviews and meetings, we had a number of informal interactions over dinner and drinks with members from both sides.

We conducted about 17 interviews in Norway (15 face-to-face and 2 over videoconference) with senior management, project leaders, developers and support staff. Also, over 10 interviews were conducted with various levels of staff at RussCo in St. Petersburg. The initial interviews in Norway were aimed at gaining a historical overview of the relationship, and subsequent meetings focused on understanding more specific details about the SalarySystem project. Some of the key respondents from the Norwegian side were interviewed more than once, sometimes to gain responses on focused issues, such as how project budgeting was done. A summary of the interviews conducted is presented in Table 1 below. Most of the interviews were conducted in a team comprising of a faculty member from the University of Oslo and two informatics masters students. Typically, an interview took place at the respondent's office, and on average lasted from 30 to 120 minutes. In these interviews, which were semi-structured in nature, we tried to obtain information about the background of the respondent, especially his or her experience with global software work, the kind of problems they experienced in the current relationship, how they tried to resolve them, and what learning they had gained from this experience. We also tried to elicit comments from the respondents on further issues which they felt we needed to explore in more detail, for example relating to how the knowledge differential between the two sides had changed during the course of the project and why.

In addition to these primary interviews, we also attended one important project meeting between the two sides in Russia (about 10 people attended) which was aimed at charting out a strategic plan for the relationship and to identify potential new directions. Attending this meeting provided us with a sense of the dynamics in the relationship, for example relating to who had more authority in making decisions. We were also provided access to a number of different documents relevant to the project, including the Software Requirements Specification (SRS), a test case, the test plan, two weekly status reports from Russia, e-mail transcripts, and several documents describing SalarySystem. These documents were especially useful for the historical reconstruction of the project, for example the SRS document provided insights into how the requirements had been understood by the Russians, who had been involved, and how this understanding was shared amongst the developers. We also analyzed in detail how *TestTool,* (a tool for managing bugs in the source code) was used, and its role in facilitating trust and control over the development process.

V. Imsland & S. Sahay • 109

Designation	Company	Number of interviews	Number of meetings
Company Manager	ScanSys	2	4
General Manager	ScanSys		3
Test Coordinator	ScanSys	3	
Developer	ScanSys	1	
Project Manager	ScanSys	1	1
Support Coordinator	ScanSys	1	
Tester III	ScanSys	1	
Development Director	ScanSys	1	3
Technology Director	ScanSys	1	2
Project Leader (on previous GSO projects)	ScanSys	1	
The Cultural Liaison		1	3
President and CEO	Indian Supplier Company	1	
General Manager	RussCo	3	1
Tester I	RussCo	1	
Tester II	RussCo	1	
Former Project Manager	RussCo	1	
Managing Director	RussCo		1
Project member	RussCo		1
Project Coordinator	RussCo	1	1
The Delphi Guru	RussCo	1	
Developer I	RussCo	1	
Developer III	RussCo	1	
Developer IV	RussCo	1	
Developer II	RussCo	1	
Project Manager	RussCo	1	
Total		**27**	**20**

Table 1. Respondent profile

Through this primary and secondary data analysis, we identified issues that we felt influenced the GSO relationship, and presented in a management report to both sides, and also over videoconference to other ScanSys staff in Oslo and Trondheim. In June 2003, we visited Russia again where we discussed the report with RussCo staff to obtain feedback which was incorpo-

rated into the analysis now presented. While these reports helped to provide the first cut to our analysis in terms of identifying practical issues influencing the relationship, a more theoretical analysis proceeded with a reading of the literature and relating this to the practical issues. For example, the practical problems identified around knowledge sharing was related to literature in IS and organization theory such as the notion of embedded knowledge. This conversation between the empirical domain and theory provides the foundation of our analysis presented in this paper.

4 Case Study

The case study is presented in a sequential manner based on how the SalarySystem project evolved historically. The first and second phases represent a historical reconstruction of the first meeting in Norway when the project was initiated and requirements were discussed. We then describe the *near-breakdown* stage when the project was nearly terminated, and the subsequent period of recovery and growth which unfolded during the course of our research.

SalarySystem had been developed in house in ScanSys since 1987 by a group of developers who were not Delphi experts, and we were told that they learnt the technology as they were developing the system. As a result, the code was not well documented and did not follow standard coding conventions with respect to comments, code structure, use of variable names etc. Then in around 2000, ScanSys went through a major reorganization, and a number of the in-house developers either were moved to different functional areas, or left the organization. The knowledge about the system and how it was developed was for all practical purposes lost, and very little memory of it remained in terms of formal documentation. Furthermore, SalarySystem was deeply connected to the rather complex and particular system of Norwegian tax and salary rules, like the special rules for sailors working abroad, foreigners working in Norway, or people living in Northern Norway. These rules were also annually updated. The Project Manager and a Russian speaking Latvian developer had both joined the project only a few months before the outsourcing project was initiated, and they too were still learning about the system when the project started. Under these circumstances, the ScanSys side was quite poorly equipped to transfer important historical knowledge about SalarySystem to RussCo.

V. Imsland & S. Sahay • 111

4.1 The First Meeting in Norway: Eliciting Requirements (2001)

After RussCo was selected as ScanSys' outsourcing partner, two of their staff (the project manager and a developer) visited Oslo for a week in 2001:

> [RussCo] appointed a project leader who came to Norway for one week to go through the specifications and learn more about our organization. Also, [the ScanSys staff] should evaluate the degree of knowledge and experience of the RussCo employees. (ScanSys consultant)

The developer was described by his colleagues to be very clever and was referred to as the *Delphi Guru*. From ScanSys, the project manager and a Russian-speaking developer of Latvian origin participated in the meeting. There were two main aims of the meeting: one, to transfer an understanding of the domain knowledge of SalarySystem to provide the basis of the requirements document (SRS) to be developed by the Russian side; and two, learning to know each other better on the social level.

We were told that the focus of the discussions was primarily technical, considering the database structure, the use of Delphi standard components etc. The ScanSys project manager told us that he had been given very little notice about this meeting by his managers, and could only hastily prepare a document that described how the new system should be. This document provided the starting point for the discussions which basically took place in Russian with the Russian-speaking ScanSys developer explaining the details. This mode of discussion in Russian contributed largely to the exclusion of the ScanSys project manager and amplified the technical approach because it became a case of primarily programmers talking to each other.

The documentation about SalarySystem was to a large extent written in Norwegian, including the comments on the source code, class names, variable names, database fields, and tables. During the meeting, the decision was made to use Norwegian in the source code of the new system too, justified by the need to maintain consistency between the old and new versions of SalarySystem which were for a period to be used simultaneously. As a result, the Russians received very limited documentation or understanding of the system requirements, especially related to the underlying business logic. The Russians were given the 15 Mb source code and were basically asked to create the same functionality on a new platform. In addition, they were provided with an installation guide, the SalarySystem manual, a photograph of the database structure from the whiteboard drawn by one of the ScanSys staff in the first meeting in Oslo, and a running version of the system. Most of the documentation was written in Norwegian.

On return to Russia, the RussCo staff on the project spent a lot of time trying to interpret the information obtained, and seek clarifications through listing questions and creation of prototypes when they could not see the business logic. A Russian developer told us:

> All additional time was spent trying to understand the business logic of the system. The business logic was not understood, and we did not see the complexity of the project. We did not understand its usage. (RussCo developer)

For example, the Russian developers and testers would compile their queries in Russian to their project manager, who would then translate it into English, and then send by e-mail to ScanSys. The same process had to be repeated back when replies were received from ScanSys. Using telephone- or video-conferences to clarify problems was also tried, but was abandoned because of language problems in addition to some technical problems in getting these conferences to work.

Since e-mails were regarded as the formal documentation of the process, a policy was made to send all messages to a mailing list in order to make visible the communication and to provide an official archive of project details over time. In this process of translation and back-translation, we were told that important meanings, especially relating to domain-specific terms were lost. For instance, employees in Norwegian companies who use their own private car at work, get some of these expenses refunded from their employer in addition to their regular salary. The use of this *bilgodtgjørelse*—translated into *car-allowance*—was difficult to understand in Russia, not to speak of trying to produce the software used to calculate it. In addition to these lost meanings, huge amounts of time were taken for this translation.

In interviews, the Russian programmers told us that sometimes there would be a delay of one up to three days to receive a reply from Norway on an e-mail query. E-mails often had to be resent for purposes of clarification which caused further delays. This led to frustration amongst the Russian team members, and they would put away a (sometimes) critical issue until a reply was received. Messages needing urgent attention where accompanied with a flag which would prompt the head of RussCo to look at that message and take action. As a result, some Russian members felt intimidated by the visibility both for the issues they raised and their relatively weak English skills.

This process of understanding was adversely affected by the fact that when the project manager on return was allocated to spend only about 10% of his time on SalarySystem, and the Delphi Guru was moved 100% to other projects. Both these internal transfers were not communicated to ScanSys who believed they were still working on the project. Queries posed by RussCo to ScanSys, especially concerning the business logic, often could not be

answered satisfactorily because the Norwegian developers themselves had only been recently appointed and had limited understanding (or documentation) about the original technical and business specifications of the system.

Despite these constraints, the Russians started to make estimates about the re-engineering of SalarySystem. Firstly, an attempt was made to estimate the system based on counting lines of source code. An estimate made based on around 300 000 lines of source code was abandoned as the estimate was seen to be too high, and also because the new system would involve a newer database structure and thus was expected to have fewer lines of code. Subsequently, a new estimation process was adopted where the source code was analyzed in relation to the running version of the system to better understand the structure and functionality. The Russian project manager then asked each project member how much time they needed to develop different parts of the system like forms, tables, reports and so on. These individual estimates were then aggregated and a further 33% was added for testing. These estimates were sent to Oslo where they were compared with the ScanSys estimates, and it was concluded by the Norwegian project manager there to be generally okay.

After the estimation process, the development work started. The Russian developers tried to work with Norwegian-Russian dictionaries to try and address the language barriers. There were a number of misunderstandings also reported. For example, one developer told us she spent 4 weeks working on a prototype based on her understanding that the new system should be identical in the functionality as the old. But when she presented her work, she was told by the Norwegian side that was not correct because in the new system the reports should not be hard coded as it was in the old. This lost time and effort caused her a great deal of frustration.

RussCo reported the progress of the project to ScanSys on a weekly basis. This was more of a formal report which the Project Manager sent, after clearance from his bosses, and it was typically reported that all was going well. No cross-verification of the reports was made either by the Russian top management or by ScanSys, as described by the Norwegian side:

> It was reported that there was no problem; [the ScanSys project manager] did not have any reason to check more thoroughly. (ScanSys company manager)

In fact, we were told that no visit from a Norwegian was made to Russia for technical discussions for nearly nine months after the project started. For the more technical queries, the developers would give their questions internally in Russian to the project manager, who would then aggregate all these questions, translate them to English, and e-mail it to ScanSys. The replies followed a similar channel backwards.

Six months after the project started, a near-breakdown stage developed which we now describe.

4.2 The Near-Breakdown

About five months into the project, ScanSys started showing signs of being displeased with the project. The software deliveries from RussCo were seen to contain lots of bugs, screens were without all the requested functionality and several important modules such as for report generation were missing. In the absence of all the modules and bugs in the existing ones, the whole system could not be tested. ScanSys started to ask RussCo some critical questions, including reasons for delays. The Russians did not agree though, claiming that everything was on schedule. Just as the project delivery date was approaching, ScanSys was told that the project was three weeks late, and then two more weeks, and finally a request for a further six months extension was received. This created panic in Norway, and a team of people flew out to Russia and a hard confrontation ensued. ScanSys blamed RussCo for the delay, pointing out while RussCo had estimated six months to finish the project, using 642 person-days, the software delivered at that time were few and the quality unsatis-fying. While this delay indicated an estimation problem, the Russians pointed out that their figures had tallied with the Norwegian figures also, and the delays were due to their incomplete understanding of the business logic, slow replies to queries from Norway, and at least 75 person-days being required for language training and understanding the business logic. But ScanSys were upset that they were not given an idea of the delays in the project till right before the delivery date, to which RussCo had no arguments against. They also admitted that they had deliberately made lower mutual estimates to gain the contract as ScanSys were there first Scandinavian customer and they were very eager to enter this perceived lucrative market.

Through this charged meeting, at least the problems came to the fore and the Russians were apologetic. The RussCo project manager was then asked to come in, and his bosses blamed him directly for the delay and publicly removed him from his post. The ScanSys staff, who had not experienced such public admonishment of staff in their own organization, were extremely sur-prised by this episode:

It was a hard situation. We are not used to it in Norway.
(ScanSys project manager)

During the meeting, the Russians offered to finish the project without being paid for additional time, and after this was agreed, additional staff including

V. Imsland & S. Sahay • 115

the Delphi Guru were deployed to the project. This started the recovery phase of the project which we now describe.

4.3 The Recovery Phase and Growth

Prior to the near-breakdown of the project, the software deliveries from RussCo contained numerous bugs. This was partly due to the Russians' lack of understanding of the business logic and with it the user needs. For example, the system of taxation for a Svalbard resident is different from that in other regions of Norway. Through this face-to-face meeting, ScanSys understood the problem of the business logic and tried to fill this gap through the use of test cases. Each such test case described a scenario or operation that was considered difficult to describe in a formal technical specification. These cases provided a very detailed description of the use of SalarySystem from the users' perspective, including what buttons to push, what values to fill in and so on. The sequence of the operations described the most common way of using SalarySystem, and in total 15 such test cases were sent to Russia after the near-breakdown situation.

Another change in the project was the introduction of TestTool; a software product which helps software projects organizing the development process. It can be used to manage bugs and feature requests as well as monitor other project related issues throughout the development cycle. A development project using TestTool has a database that can be accessed by all assigned project members. TestTool also has a web-based client, which makes it suitable to support virtual teams since the database can be accessed from wherever there is an Internet connection. Almost universally, members from both sides described that the use of TestTool helped to make visible the problems, and provided the Norwegians with transparency and thus better control over the project. The weekly reporting system was also improved with the top management personally verifying the progress reports being sent to Norway and the Norwegians asking more direct questions.

Three months after the near-breakdown, three ScanSys staff; the project manager, the test coordinator, and a developer went to Russia, representing the first face-to-face meeting (9 months after the project was initiated) on technical matters. During this meeting, the Norwegians explained both the technical details about SalarySystem and also the business logic concerning Norwegian tax and salary rules. Through the meeting, the Russians reported that they came to better understand the calculation engine that was being developed in Norway. Since the calculation engine was linked to the module being developed by the Russians, the hitherto absence of understanding of that module

116 • V. Imsland & S. Sahay

had also contributed to the delays till date. Both in Norway and Russia, the meeting was considered very positive, although it was also generally agreed that such a meeting should have taken place earlier:

> If this kind of meeting had been held earlier in the project, it would have changed the project dramatically. (The Delphi Guru)

After this meeting, more face-to-face meetings took place both in Norway and Russia which also helped to develop a more healthy social relationship. Some of the Russian developers expressed that they had developed a passion to learn more about Norway and made independent efforts to learn the Norwegian language and culture. Some of them told us they would like to visit Norway on vacations and experience the unique nature there.

We were also told that ICQ (an instant messaging system) was started to be used by the Russian developers to communicate with the Russian speaking Norwegian developer which enabled a direct and immediate interaction to solve technical queries. This system provided the Russian developers with rapid responses to their queries. In addition, the ICQ represented a *safe haven* as compared to the earlier cumbersome system of e-mails that was not frequently used by them because of language inhibitions and also since the message went to a large mailing list including their top bosses.

In summary, the recovery of the project was firstly bought about by both sides agreeing to the mistakes made in the earlier phases and mutually agreeing to learn from them and improve future operations. These improvements were facilitated through a number of measures such as increased face-to-face meetings, use of test cases and TestTool, improved reporting and the locally evolved use of ICQ. Another interesting dynamic concerned the shift in the knowledge base around SalarySystem. When the project started, ScanSys were in a position of strength since they had several software companies to choose from while RussCo was the weaker partner trying to impress. The project contract, signed two months after the project start, illustrates this asymmetry. According to this contract, the project was divided into eight phases where the eighth phase was delivery of the pilot version. Each phase had a maximum number of man hours, and if this work amount was exceeded, RussCo had to pay for the rest of the phase. If the job was completed in less time, the unused man-hours could be carried over to the next phase. Also, both companies had the possibility to stop the project after any finished phase. 14 months after project start, when the amount of work had exceeded the estimates by more than six times, the project was still in phase seven. ScanSys then had the choice of stopping the project there, thereby getting the SalarySystem software at low cost. However, ScanSys by now were dependent on RussCo who had by then gained a good technical understanding of the system.

V. Imsland & S. Sahay • 117

This dependency was magnified by the fact that ScanSys, in the process of restructuring their own organization, did not have access to significant developer resources.

This shift in knowledge dynamics, coupled with ScanSys' recognition of the basically good quality (and low cost) of RussCo's developers, made them decide to finish phase eight (of testing) internally, but still pay RussCo $14 000 according to the original contract. This was to compensate RussCo for the extra resources that they had committed earlier on their own costs. ScanSys also saw this compensation as a valuable investment to ensure a healthy future cooperation for further projects. In a recent meeting with the Norwegian senior manager to assess the overall outcomes of the project, we were told that the results had been mixed. At the project level, more than three years after the SalarySystem project was launched, the product still suffered from poor quality. We were told that while the new database was technically extremely well developed and the best ever, the user application on top of it still had numerous bugs. As a result, a full enrolment of all customers was still to come, and significant delays had been experienced.

However, at the level of the relationship, the manager said that significant learning had taken place for both sides about how to perform better GSO projects. A concrete example of this learning could be gained from the fact that following the SalarySystem project, a number of other projects had been taken with the Russian firm and average cost savings experienced had been on 35-40% in average, and the software was also seen to be of a better quality technically as compared to what would have been developed in Norway. This experience has even prompted the Norwegian firm to consider acquiring part of the Russian software house to support their future strategy of scaling up their outsourcing operations in Russia. Organizationally, both sides had made a number of improvements from trying to establish a flatter hierarchy in Russia and to enable more direct lines of communication between the development teams from both sides. The Russian project members had made conscious attempts to improve their language skills, and were also encouraged to report the real project status. In Norway, the project managers started to ask for CVs from the Russian developers involved in new projects so as to get a sense of their competence, and sometimes even interviews were conducted. Despite the increased costs, the frequency on face-to-face meetings on projects was deliberately enhanced.

The overall experience had contributed to the creation of a new competence in both sides on how to manage GSO projects, and the SalarySystem project, despite its obvious problems, had contributed to a larger success by providing a basis for organizational learning.

5 Case Analysis

In this section where the case analysis is presented, two key questions are examined: what is the nature of the complexity around managing knowledge in this project and how were they embedded; how can this complexity be better addressed, including what were the formal and informal processes of negotiation and appropriation from both sides to make these knowledge systems *less embedded* or *more migratory*?

5.1 Complexity of Knowledge Systems

Software development, in general, brings into play varying kinds of knowledge systems. This variety is further magnified in the context of GSO because of the diversity of people, organizations and countries that are involved in the process. In the case described, we argue that the complexity of the knowledge systems arise from the multiplicity of understandings required including those related to *domain knowledge*, *language*, and *project management*, and how these need to be integrated to give overall direction to the project. We first discuss each of these three knowledge systems independently, followed by why they need to be integrated and the underlying challenges.

Domain knowledge. The term domain knowledge has been used to primarily refer to the application area of payroll management in Norway. As described in the case, the SalarySystem project concerned the redesign of a payroll management system from an existing platform to a new one. This development process brought into play the need for different understandings ranging from about the rules, regulations and legislation governing the salary structure in Norway, to the technical details about the system including its database structure, meaning of variable names in the source code, estimation methodologies, and technical platforms on which the software was originally built and also being migrated to. At the start of the project, such domain knowledge was rather restricted within the Norwegian company and nearly non-existent on the Russian side except to the extent of some expertise in Delphi programming more generally.

The case analysis identified that domain related knowledge was embedded at multiple levels of society, organization, and project. At the societal level, domain knowledge was embedded within the salary and tax rules that were commonly used in Norwegian institutions, and thus taken for granted there, but was largely unknown and not communicated effectively to the Russian developers who needed to understand the business logic to effectively develop the system. At the organizational level, domain knowledge was embedded

V. Imsland & S. Sahay • 119

within the organizational history, of how the original SalarySystem had been developed, by whom, the ad hoc manner of development, the lack of documentation, and how this knowledge had been largely lost because of the people being moved out. At the project level, knowledge was embedded within the particular practices of project management, for example in not involving a business analysis in the requirements meeting.

As a result of the above, in addition to the thin knowledge about the technical details of the system, the business related perspective of how the system would be used by firms was also largely missing. As in the case of the technical side, the business or product managers relating to the SalarySystem had also been relocated in the organization, and there was nearly no one the technical people could access to gain an understanding of use related issues. There was no business analyst involved in the discussions between the Norwegian and Russian around requirements, and the meeting took on a very technical focus, also compounded by the fact that the Russian representatives in the requirements meeting were technical by training and experience, and because of language issues since it was primarily the Russian speaking Latvian programmer who was involved.

These issues of technical focus, lack of business analysis, and complexity of the Norwegian salary and tax rules all contributed to the Russians not having adequate knowledge about the business logic underlying SalarySystem. This embeddedness of knowledge at these three multiple and inter-connected levels made the understandings of what was the SalarySystem and its transmission to the other side largely non-migratory. What little knowledge that had been gained by the Russians was further dissipated by the movement of the two Russian staff who had attended the Oslo meeting to other projects. Since this movement was not communicated to the Norwegian side, they were blissfully unaware of the knowledge gap that had been created. The absence of any visits by the Norwegians to RussCo meant that the lack of this knowledge was unknown to ScanSys, until too late. All these issues led to serious challenges in carrying out the development work, both with respect to the content of the system, and within the structure of the time and cost constraints of the project.

Language. In GSO relationships more generally, for example involving North American (and Indian firms), the medium of instruction is typically English. Proficiency of a large technical workforce in English language is one of the well cited reasons for the success of the Indian outsourcing industry (Arora et. al. 2001). As outsourcing is being extended to different geographical domains like Europe and East Asia, language becomes a crucial issue, and for example in India, firms are investing significantly into language and cul-

tural training. Sahay et al. (2003) report of an Indian firm doing work in Japan who hired Japanese anthropologists for an extended period to train the Indian developers, not only in the Japanese language, but also in Japanese culture, rituals, and stories. This approach reflects an understanding of how language related issues are more than just language but are deeply embedded in societal conditions of culture, rituals, everyday life, and belief systems.

The SalarySystem from the perspective of language was unique in that both sides were trying to communicate in a common language (English) that was not their native one (Russian and Norwegian). While the formal project policy was that all communication would take place in English, in practice three languages were involved – English, Russian, and Norwegian. While Norwegian was the language used internally in ScanSys, in RussCo, Russian was spoken between the staff. Relative to the Norwegians, the command over the English language of the Russians was very weak and led to all communication from and to the Russian side being channelled through the moderately proficient Russian project manager. Because of the multiplicity of languages being used, a number of translations were needed to be made.

Language issues were thus both embedded at the societal level of the languages that the different groups had grown up using, and also at the organization and project levels. At the project level, the embeddedness came from the fact that the entire documentation of SalarySystem provided to the Russians was in Norwegian, including variable names, the user interface, and also the comments on the code. This project level condition was further magnified by the organizational level practice of the Russians arising from their respect for hierarchy and the principle that that the customer is always correct. As a result, the Russians did not explicitly articulate they could not understand Norwegian, and instead tried to use a dictionary to try and muddle through. This was not easy given the technical nature and socially embedded nature of some of the terminology used. The literal translation of some of the technical terms was not easily found in the Norwegian-Russian dictionary being used by the Russians, and remained thus largely non-migratory. The car-allowance-related case referred to above provides such an example.

Project Management. We use the term project management knowledge to refer to understanding and communicating about the progress of a project with respect to milestones, deadlines, and budgets. Effective knowledge about project management can help managers to effectively monitor the project status and take relevant action when required, for example provide more resources, re-consider estimates, and relocate budgets. The SalarySystem project was characterized by inadequate knowledge about project status and

V. Imsland & S. Sahay • 121

progress. This inadequacy can be seen to be again embedded at the levels of society, organization and project.

The societal level embeddedness can be seen to arise from the general principles of governance in Norway which are based on trust. So, when someone tells that he or she will do this or has done that, the statement is normally taken on face value and trusted. For example, in the SalarySystem project, the reporting was done on a weekly basis. We were told that this report was sent by the RussCo project manager, and it was quite general in nature and primarily stating that everything was going OK. This report was generally taken on face value by the Norwegians and they believed that everything was indeed okay.

The problem of this societal level form of trust was further compounded by the organizational principle of hierarchy in Russia where the project manager could not communicate anything directly without getting the OK from his bosses. The Russian top management, we subsequently learnt, had rarely ever checked the veracity of the project manager's reports by physically examining the state of the activities being reported as OK. Similarly, the Norwegians never cross-checked on the reports and took them on face value. As the project progressed, and reports were sent weekly, the practice of sending and receiving reports without verification from both ends became institutionalized. This practice was only brought to question with the creation of the near-breakdown situation.

Project related knowledge was further embedded by certain project management practices. For example, we were surprised to hear that the Norwegians did not have any information on the background of the developers who were working on the project. A relatively standard practice amongst North American firms doing outsourcing is to request for the CVs of developers working on the project to know their names, experience and skills. Not only did the Norwegians not have this information about the developers, they also were in the dark about the fact that the two key Russian staff who were supposed to have the knowledge about the system had been moved to other projects.

Reporting on projects in GSO settings need to be sensitive to the different contexts and varying systems of work. For example, the Russian management style was more hierarchical as compared to the ScanSys approach. An implication of this was that the Russian project manager would not feel comfortable enough to report on the real state of the project and the problems being experienced without getting the approval of his bosses, who had their own reasons to project that there were no problems being experienced. The Norwegian managers, used to a system of trust and egalitarian style of working, took the

project reports on face value as they believed they were being informed the real status. The shock which the Norwegians felt when they saw the Russian project manager being publicly shamed and reprimanded during the meeting in Russia also reflects the very different styles in dealing with employees and approaches to hierarchy. From the Russian perspective, they wanted the Norwegians to be more proactive when it came to sharing the domain knowledge about SalarySystem or in replying to their queries, but were not used to questioning the customer, seen as being superior in the hierarchy, within their culture and working practices. Furthermore, the Norwegian style of working caused delays because of slow feedback on questions and generally low interaction on the details of the project level. The absence of face-to-face contact for nearly nine months into the project, again a relatively standard practice in North American outsourcing arrangements, compounded the existing problems.

In the absence of face-to-face contact, the technical problems in getting the video-conference to work, and the problem of English language conversation over the phone meant that e-mail became the primary and also official mode of communication, especially for project management purposes. But due to the potential it provided for increased control through visibility, some Russian members felt reluctant to post messages to this list, and instead relied on informal and internal communication like through the use of ICQ in Russian language. A significant proportion of important project related documentation thus remained invisible to the broader project team on both sides.

In summary, we have discussed a variety of issues around knowledge, how they were embedded at multiple levels of society, organization and project management conditions. This embeddedness contributed to challenges in sharing knowledge, with direct implications on the progress and outcomes of the project. In the table below, some of these issues are summarized, following which we discuss the negotiation and improvisation process through which the actors tried to deal with the knowledge related challenges.

Actors from both sides attempted to deal with the multiplicity of knowledge systems, its embededdness, and the gaps they created in understandings. There were informal methods employed like the Russians studying the Norwegian-Russian dictionary to interpret meanings of words and with it deal with the challenges arising from the embededdness of language and the knowledge gaps they created. A number of formal methods involving technology were also initiated as a result of the near-breakdown situation described in the case. This situation served as a benchmark event which served to catalyze various learning processes to develop improved shared understandings and reduce the knowledge gaps between the two sides. We discuss four key proc-

V. Imsland & S. Sahay • 123

Problematic areas around knowledge	Where Embedded	Nature of Embedding	Implications on Project
Domain related	The system of Norwegian tax and salary rules. The history of ScanSys and how SalarySystem had evolved in the organization. The Russian system of technical education and also business where business related issues are deemphasized and also poorly understood.	Tax rules deeply embedded into the business logic of SalarySystem. Business logic understood by few people at ScanSys who were now no longer in the organization, and not was there remaining documentation.	Caused delays because the Russians lacked this knowledge at the beginning of the project, and the Norwegians failed to transfer it. Lack of proper understanding led to systems being specified incorrectly or prototypes being developed which were later described as not being needed.
Language related	The respective national systems of education – Russian, and Norwegian. Limited access to English language education from the Russian side. The limited experience of the Norwegian side to work outside Scandinavia, thus assuming that Norwegian language in SalarySystem would not represent a problem when outsourcing the redesign to Russia.	English used as official language although none of the project members had English as their first language. All relevant documentation provided to Russians in Norwegian language with little support from ScanSys for their translation and interpretation.	Caused misunderstandings in interpreting Norwegian text, especially relating to technical terminology. Due to long time spent on translation, overheads were created, which had not been accounted for in the original estimates.
Project management related	The Russian organizational practice where hierarchy was emphasized. The Norwegian relatively flat and egalitarian style founded largely on trust.	The top management at both sides failed to understand the project status prior to the near-breakdown. No face-to-face contact, especially for developing shared understanding on technical requirements and the associated business logic.	Kept the top management from taking relevant action to improve the transfer of domain knowledge, language skills etc. Limited shared understanding of the needs and progress of the project. Led to a near-breakdown situation, which however had its positive effect of making problems visible and emphasized the need for corrective action.

Table 2. Managing the complexity of knowledge through negotiations.

esses of negotiations that were central to dealing with the embeddedness of knowledge: *use of TestTool*; *use of Test cases*; *use of ICQ*; and *increased face-to-face interactions*.

Use of TestTool. Because of the distance, the limited face to face meetings, and the prior experience leading to the near breakdown situation, the Norwegians felt a huge knowledge gap about their understanding of the state of the project and its progress. There was the need to make visible over distance, what was happening on the Russian side, for example, what are the existing bugs, who is responsible, what is the status of its being fixed etc. To deal with this knowledge gap, TestTool was introduced into the project on the initiative of the Russian side after the near-breakdown situation, and helped the Norwegians to develop trust because of increased visibility of operations relating to the registration, fixing, monitoring, and reporting of bugs. Project related knowledge which till date had been largely embedded because of the project management practices and organizational structures, through the use of this tool became relatively more visible to both sides.

Use of Test cases. Test cases help to simulate a use scenario, which contributed to improve the sharing of domain knowledge from Norway to Russia. The most positive effect of these was that through them, the Russians learnt how to use SalarySystem themselves. As they started to play around with the system and work with it, they started to understand some of the business logic and the uses perspective. Before this, SalarySystem, for the Russians had been just a piece of code, and now slowly it started to be seen as an application being used by people. This shift in perspective and increased knowledge helped the Russians significantly in the redesign process. The use of test cases also contributed to improve communication between the two sides by creating a common framework within which the discussions could take place around both the technical and use issues. The use of these cases as a communication medium further contributed to build up and sharing domain knowledge in the SalarySystem project. Embededdness of knowledge, arising as a result of domain and business understanding, was partly addressed through the use cases in two ways. Firstly, by providing concrete examples of the domain which helped the Russians to go underneath the code and see the application world. Secondly, by providing a channel for communication, more questions could be asked and understood relating to the domain.

Use of ICQ. The use of ICQ evolved at an individual level with some of the Russian developers and the Russian speaking Norwegian staff installing the system on their machines. In addition to be able to get rapid responses to their queries, they were able to intersperse small bits of personal information with each other along with the technical. This interaction helped to develop both

V. Imsland & S. Sahay • 125

the technical and social side of the relationship. There was however some negative implications of the ICQ, especially from the perspective of the Norwegian Russian speaking developer. As the Russians started to realize the effectiveness of this medium of communication, the number of queries from them started to increase. Slowly, the Norwegian found that he was being inundated with questions, and there were expectations from the Russians for immediate responses. He found dealing with these questions was also interfering with his ongoing work for which he was responsible to his management, and he was not being appreciated for dealing with this invisible work. Interestingly, the key role that ICQ was playing in the project management was totally oblivious to the top management from both sides. When we were making the presentation of our findings to the Norwegian management we were extremely surprised to hear that they had decided to de-install ICQ from the ScanSys systems because of security reasons.

In summary, the use of ICQ played an important role in negotiating some of the embedded knowledge about the technical details of the system arising from issues of language, weak historical understanding, and the hierarchical structure of working of the Russian organization.

Increased face-to-face interaction. Initially, nearly nine months into the project, there had been no face-to-face interactions between the technical people apart from the first meeting in Oslo. As a result, a significant proportion of the domain knowledge remained embedded in Norway, and especially inaccessible to the Russian developers. This embededdness was further magnified because of the lack of face-to-face meetings. The absence of use of videoconference, very limited use of telephone, meant that the primary source of communication was through e-mail, which again was not always effective because of language difficulties and the delays in getting replies (from both sides). The near-breakdown situation forced the initiation of face-to-face contact when the Norwegian team rushed to Russia to try and salvage the crisis situation. This meeting could be seen as a turning point in the relationship, bringing to light the problems and emphasizing the need for more face-to-face contact for improving technical knowledge and to help develop a social relationship as both sides were able to put faces to names of people they had been communicating with. This helped both sides to develop a sense of comfort in their communication and more realistic expectations about the replies they could expect both in terms of content and frequency. The discussions that followed during the meetings also helped to establish more consistent project management routines.

In summary, the event of the near-breakdown situation helped to trigger the realization of the need for face-to-face meetings, which on one hand helped to

unpack both the technical and social understanding embedded in both sides, and also to develop improved rhythms for interaction which had been largely lacking in the past.

6 Discussions and Conclusions

In this paper, we have provided an in-depth empirical analysis of a relationship between a Norwegian and Russian firm engaged in the process of global software development. The paper makes a unique contribution in providing an inductive analysis of knowledge related issues salient in managing such a relationship in the particular context of a Norwegian-Russian relationship.

Theoretically, the paper extends the idea of embedded knowledge proposed by Nicholson and Sahay (2004) developed based on their analysis of a British-Indian relationship. Through this inductive analysis, we identify three additional issues that contribute to the embededdness of knowledge. The first concerns language, as two groups try to communicate with each other in a language that is not their first language. The issue here is not only about the literal problems of translation but about the meaning of words and how they relate to particular contextual conditions (the example of car-allowance provided above). A second issue discussed is about how knowledge is embedded in the software itself, for example in our case of how the business logic remained invisible in the code—for reasons both technical and institutional. A third issue concerns project management practices, and how the varying cultural ways of working (for example related to different hierarchical systems in Russia and Norway) contributed to knowledge about the project management status being embedded.

We also go beyond Nicholson and Sahay in discussing how actors try to deal with this embedded knowledge in locally improvised and negotiated ways such as through the use of ICQ. Events, such as the near-breakdown situation, also provide the impetus for making visible the problems caused by embedded knowledge, and force both sides to look for appropriate solutions to deal with them. These solutions came in the form of use of the TestTool, test cases, and increasing face-to-face interaction. A key insight that arises from our analysis is how these locally or (more globally) shaped efforts contributes to learning and reflexively making improvements to ongoing efforts. Given the long term nature of a global software development relationship, nurturing such local and global learning processes is seen as absolutely critical for survival.

V. Imsland & S. Sahay • 127

References

Abbot, Y.P., "Proximity-Based Models of Offshore Software Outsourcing: Exploring the Concept of Location in Nearshore and Onshore Software Outsourcing Ventures," Unpublished PhD thesis, Cambridge, UK: Cambridge, 2004.

Arora, A., Arunachalam, V., Asundi, J., and Fernandes, R., "The Indian software services industry," *Research Policy*, (30), 2001, pp. 1267-1287.

Badaracco, J., *The Knowledge Link*, Harvard Business School Press, Harvard, 1991.

Bechky, B., "Sharing meaning across occupational communities: the transformation of understanding on a production floor," *Organization Science*, (14:3), 2003, pp. 312-330.

Carmel, E., *Global Teams: Collaborating Across Borders and Time Zones,* Upper Saddle River, NJ: Prentice Hall, 1999.

Carmel, E., "Introduction to the special issue: The emergence of software exporting industries in dozens of developing and emerging economies," *The Electronic Journal of Information Systems in Developing Countries (EJISDC),* (13:1), 2003, pp. 1-2.

Castells, M., *The Internet Galaxy – Reflections on the Internet, Business, and Society*, Oxford University Press, Oxford, 2001.

Cockburn, A., *Agile Software Development*, Addison Wesley Press, Boston, 2002.

Cramton, C., "The mutual knowledge problem and its consequences for dispersed collaboration," *Organization Science*, (12:3), 2001, pp. 346-371.

Curtis, C., Kransner, H., and Iscoe, N., "A field study of the software design process for large systems," *Communications of the ACM*, (12:3), 1988, pp. 346-371.

Hall, E.T., *The Dance of Life: The Other Dimension of Time*, Anchor Press/Doubleday, Garden City, New York, 1984.

von Hippel, E., "Sticky information and the locus of problem solving: implications for innovation," *Management Science*, (40:4), 1994, pp. 429-439.

Imsland, V., Sahay, S., and Wartiainen, Y., "Key issues in managing a Global Software Outsourcing relationship between a Norwegian and Russian firm: Some Practical Implications," *Proceedings of IRIS26*, Haikko, Finland, 2003.

Karnik, K., "US cannot stop outsourcing," 2004, a press statement carried in http://www.rediff.com/money/2004/sep/13bpo.htm, last accessed June 2005.

Krishna, S., Sahay, S., and Walsham, G., "Managing cross-cultural issues in global software outsourcing," *Communications of the ACM*, (47:4), 2004, pp. 62-66.

Krishna, S., and Sahay, S., "GSO experiences in Korea and Japan: Some preliminary investigations," final report for the project *The Context of Innovation of the Information Technology Industry*, of University of Pennsylvania, Institute for the Advanced Study of India, New Delhi, 2001.

Lam, A., "Embedded firms, embedded knowledge: problems of collaboration and knowledge transfer in global cooperative ventures," *Organization Studies*, (18:6), 1997, pp. 973-996.

Lam, A., "Tacit knowledge, organizational learning and societal institutions: an integrated framework," *Organization Studies*, (21:3), 2000, pp. 487-511.

Mandag Morgen, "Norske bedrifter vil flytte ut 80.000 arbeidsplasser," Mandag Morgen # 30, October 4, 2004.

Maznevski, M.L., and Chudoba, K.M., "Bridging space over time: Global virtual team dynamics and effectiveness," *Organization Science*, (11:5), 2000, pp. 473-92.

Nance, K.L., and Strohmaier, M., "Ethical Accountability in the Cyberspace," *Proceedings of the conference on Ethics in the computer age*, ACM Press, 1994, pp. 115-118.

Nicholson, B., *The Process of Software Development Across Time and Space: The Case of Outsourcing to India*, Unpublished PhD thesis, Manchester, UK: Salford University, 1999.

Nicholson, B., and Sahay, S., "Embedded knowledge and offshore software development," *Information and Organization*, (14:4), 2004, pp. 329-365.

Robey, D., Khoo, S., and Powers, C., "Situated learning in cross functional virtual teams," *IEEE Transactions of Professional Communication*, (43), 2000, pp. 51-66.

Sahay, S., Krishna, S., and Nicholson, B., *Global IT Outsourcing: Management of Software Development Projects*, Cambridge University Press, Cambridge, 2003.

Sarker, S. and Sahay, S., "Implications of Space and Time for Distributed Work: An Interpretive Study of US-Norwegian Systems Development Teams," *European Journal of Information Systems*, (13:1), 2004, pp. 3-20.

Szulanski, G., "Exploring internal stickiness: impediments to the transfer of best practice within the firm," *Strategic Management Journal*, (17), 1996, pp. 27-43.

Verdens Gang, "Flagget ut til India," Verdens Gang, May 31, 2004.

V. Imsland & S. Sahay • 129

130 • V. Imsland & S. Sahay