

Knowledge Management in Distributed Software Development Teams – Does Culture Matter?

Alexander Boden
University of Siegen
alexander.boden@uni-
siegen.de

Gabriela Avram
University of Limerick
gabriela.avram@ul.ie

Liam Bannon
University of Limerick
liam.bannon@ul.ie

Volker Wulf
University of Siegen
volker.wulf@uni-
siegen.de

Abstract

Software development requires complex context specific knowledge regarding the particularities of different technologies, the potential of existing software and the needs and expectations of the users. Hence, efficient knowledge management counts amongst the most important challenges for software teams. In international teams, one of the most important issues regarding knowledge sharing is the impact of culture under different aspects: national, organizational and professional. There seem to be very few studies dealing with the issue of culture in regard to knowledge management practices in GSE. We want to contribute to the discussion by presenting case studies of small size software teams dealing with international software development in the context of offshoring. In doing so, we illustrate how cultural and social issues influence the way knowledge exchange is performed by analyzing several practices of knowledge management, considering the role of artifacts and tools, of meetings, knowledge brokers and mutual visits between sites.

1. Introduction

Software development is a creative and knowledge intensive practice. Much work in software development involves customizing a product according to the distinct requirements of specific customers, making each software project more or less unique. Amongst others, software development requires complex and context specific knowledge regarding the particularities of different technologies, the potential of existing software and the needs and expectations of the users, as well as a great deal of creativity in regard to reaching the project aims in given time and within its budget.

Good knowledge management (KM) practices are a major success factor for software development, influencing software quality and team performance. Being a challenge even for co-located teams, KM can get much more difficult in global contexts. International teams have to cope with a multiplicity of organizational, temporal, spatial, legal, national and cultural barriers, which can affect the development pace and the quality of the software. Much has been written on strategies of dealing with these barriers in the context of Global Software Engineering (GSE). With regard to KM, most approaches in the literature deal with “canonical” concepts of knowledge as a product—suggesting that knowledge can be de-contextualized and shared explicitly amongst teams relying on databases and ICT [1]. However, this “knowledge as a product” view is questionable. Practice-based approaches and theories of social learning [cf. 2] suggest that while ICT may be well suited for dealing with explicit knowledge, implicit knowledge cannot be shared out of context. Hence, these alternative approaches are focusing on understanding how knowledge is embedded in social work practices and how actors actually share and put their knowledge to practice [3].

One important issue with international teams is the impact of culture—in all its aspects: national, organizational, and professional. While the topic of “culture” is one that has interested the SE community for some time [4, 5] most of this work has tended to focus on attempts to apply, for instance, Hofstede’s [6] work on dimensions of national cultures in what we believe to be problematic ways. There seem to be very few studies dealing with the issue of culture in regard to practice-based knowledge management in GSE. We want to contribute to the discussion by presenting case studies of small-sized software teams dealing with international software development in the context of

offshoring. In doing so, we want to illustrate how companies deal with knowledge exchange in practice, and how cultural influences (in a broad sense) affect knowledge management practices, in the particular case of small and medium-sized enterprises.

The paper is organized as follows: after a discussion of the related literature (section 2) we present our cases (section 3) as well as our methodology (section 4). Then, we present our findings (section 5) and discuss the data in relation to our research question as well as the existing literature on this topic (section 6) before concluding in section (7).

2. Related Work

2.1. Knowledge Management in (Global) Software Engineering

Knowledge Management (KM) is “a method that simplifies the process of sharing, distributing, creating, capturing and understanding of a company's knowledge“ [7]. KM as a field is interdisciplinary and involves a wide range of theories and research methods. In regard to Software Engineering, there seems to be a focus on *technocratic* and *behavioral* approaches, although technocratic approaches clearly dominating the scene [1]. This focus on rather traditional knowledge management concepts is kind of problematic, as it supports a view which considers knowledge as being a possession that can be de-contextualized, captured, and disseminated without a loss of meaning through information systems. While this approach may have limited applicability for traditional software engineering concepts, the growing field of agile development requires different knowledge management strategies with a stronger focus on knowing-in-action.

Agile development methods have a growing impact on software development organizations, especially in the case of small enterprises. Agile methods propose different ways of dealing with knowledge, including less documentation and codification, while focusing on social team interaction and customer collaboration [8]. Hence, knowledge is rather thought of as being socially embedded, and appropriate strategies have to consider social, cultural and practice-related aspects of knowledge management [9]. This is reflected in a broad set of theories which propose that action is situated [10] and deeply connected to tacit knowledge [11], which can not (or only partially) be made explicit. Huysman and de Wit [12] have labeled this transition to tacit and emergent aspects of knowledge as the ‘second wave’ of KM. In this socio-technical

understanding of KM, the focus shifted from setting up canonic knowledge databases to supporting informal knowledge sharing of communities by tools which are grounded in the practices of the particular fields [13].

Granovetter [16] has emphasized the role of social connections for the functioning of organizations. Similarly, the concept of “social capital refers to network ties of goodwill, mutual support, shared language, shared norms, social trust, and a sense of mutual obligation that people can derive value from“ [17]. It can be understood as a form of social glue holding together communities, and has been emphasized in its positive aspects of promoting voluntaristic behavior in communities and supporting knowledge management of organizations [18]. According to that, members of communities (for example teams within organizations) with high levels of social capital will have a higher motivation to share their knowledge, thus implying long term benefits for the organization as a whole [12]. However, social capital can also have negative effects as it may lead to conflicts [19] or dysfunctional behavior of communities, if for example subgroups with high levels of social capital refuse to cooperate with other members of the company.

Furthermore, in distributed settings it can be very challenging to deal with the related organizational, temporal, spatial, legal, national and cultural barriers. In regard to knowledge management in distributed teams, Milewski et al. [14] have presented the concept of *bridges* from a social network perspective. In their view, human actors play key roles in social networks, influencing the fate of software development projects. Different sources name these people ‘information brokers’, ‘boundary spanners’, ‘gatekeepers’, or ‘cultural liaisons’ [15]. These roles are usually not bestowed formally, although their importance has been noticed both by practitioners and researchers. Rather, bridges are facilitated by people who manage communication and fill the structural holes in social networks. Usually they work across boundaries, visiting remote sites and spending time working there, or they are expatriates who have lived in different countries and experienced different cultures. These knowledge brokers rely heavily on their own social skills and on the social relationships they build in time. Their contribution becomes more important when teams are confronted with unusual challenges, like in the ever-changing field of software development. Regarding the role played in bridging the communication between sites, they usually act as facilitators, but they can also, on occasion, become

bottlenecks if all communication is channeled through them.

2.2 Cross-cultural aspects of Global Software Engineering

Cultural compatibility is often described as an important factor in determining the success of international software development teams. The impact of culture on software development—be it national, or organizational culture—is a topic with long tradition in Information Systems research. The recent spread of global development teams has spurred interest in this topic and led to a broad variety of studies investigating the impact of cultural issues on ICT adoption, use, and development [20].

The cultural terms used in the GSE literature often focus on national aspects of intercultural work [21-23]. These approaches usually treat culture as equivalent to national identity, referring to Hofstede's framework of cultural dimensions [6]. Within the organizational studies field, Hofstede's formulations have been the subject of extensive critique (for an example, see [24]). Criticisms of this approach include 1.) Culture is seen as a never changing, monolithic concept. 2.) Cultural groups are seen as homogeneous, while the possibility of diverging subcultures is ignored. 3.) Actors are allocated to one culture at a time [25], while different cultures are seen as being mutually exclusive [20]. The wholesale adoption of this approach by certain software engineering researchers probably has more to do with the relatively straightforward way these concepts can be operationalized and data "captured" using easy-to-apply survey instruments, than to any real engagement with the underlying organizational "theory".

Critiques of this reification of national cultures has led to a number of alternate accounts of culture. Some focus attention on the many different forms of "culture"—professional, organizational, etc.—that may affect local practices. Others develop more nuanced interpretations of the culture concept itself—moving from a focus on the concept as denoting a set of pre-programmed stereotypical behavioral responses to an understanding of the dynamics of interaction within and across professional, organizational and national boundaries. In this view, in order to obtain a rich understanding of cultural influences of knowledge management in global teams, it is necessary to investigate actual work practices in their social (and cultural) embedding.

According to interpretivist approaches, we see culture as a reference framework, which stipulates roles

and interpretations, and which is dynamically negotiated by the actors in the course of their daily work. This understanding of culture entails many different layers referring to national, professional, or religious aspects, which are seen as being intertwined in a complex, non-hierarchic way, and which can hardly be studied in isolation [20]. It also includes many invisible aspects which cannot be studied directly, like values, beliefs, and attitudes. However, it is possible to study culture by referring to its manifestations in form of artifacts, practices, and routines, which will be in the center of our attention. Hence, we are more interested in the actors' interpretations and related processes of sensemaking, than in the definition of cultural particularities [26].

We believe that this approach offers a much more holistic understanding of culture, bridging across national, organizational and professional aspects. Hence, it can be a useful lens for researching into the complex interrelations between knowledge management and international software development work. We show how we have attempted to apply such an approach in our case studies described below.

3. Cases

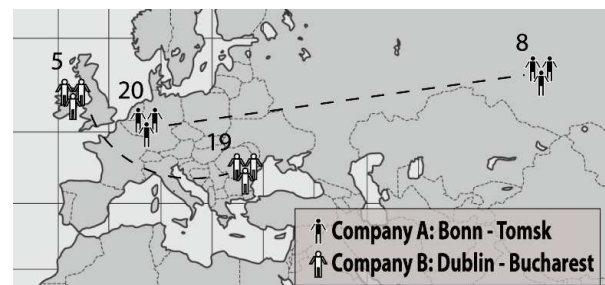


Figure 1. The locations of the teams. Numbers indicate the team sizes [27].

3.1. Germany (Bonn) – Russia (Tomsk)

Company A is a small German software enterprise engaged in the field of statistics and documentation. The customers are mainly German archives and museums. The company was established in 1980 in Bonn and has approximately 20 employees. In the mid-1990s the company found it increasingly difficult to hire German developers, as wages had increased considerably and the labor market shrunk. Hence, based on a positive experience with a very talented Russian developer who did an internship with the company, the owner of company A decided to expand his company to Russia and founded a branch in Tomsk, Siberia. Since then, an average of four to eight

employees are working for company A in Tomsk, including the former intern. The first project aimed at reengineering an existing product, which had to be rebuilt in C++. Hence, despite considerable delays in development, offshoring enabled the company to redesign their existing products based on a modern architecture. This created a competitive advantage for the company that would have been impossible to acquire otherwise, as competent C++ programmers were far too expensive in Germany at that time. As a result, the offshoring cooperation was expanded to several small size projects, using Russian developers working in close cooperation with a German project manager in customizing software products to the special needs of particular customers. Recently, the company has attempted entering the Russian market by also acquiring Russian customers. Despite the exceptionally long lasting cooperation (more than ten years), the company is still concerned with daily problems of inter-site cooperation which will be described in the following sections.

3.2. Ireland (Dublin) - Romania (Bucharest)

Company B was established in January 2006 in Dublin, Ireland. The two owners had worked together for four years in a company providing software applications for telecoms and media companies. During that period, one of them had been a project manager and the other (originally from Romania) had been working on his team as senior developer. In January 2006, they decided to leave their employment and set up their own company. They hired 4 developers in Ireland to work on their first project, and they took on project management positions. In an attempt to acquire other customers and expand the company, they tried to recruit new developers in Ireland, but failed due to the harsh competition. Consequently, the Romanian project manager identified a small company with five employees in Bucharest, Romania which they acquired. The Romanian company is legally independent and incorporated in Romania, but the same two managers, (Irish and Romanian), have equal shares in it. In December 2007, there were 19 people working in the company's offices in Romania, and another project manager (besides the Irish manager) in Dublin, and the Romanian manager was traveling between Dublin and Bucharest frequently. In January 2009, the number of employees had grown to 26, of which 7 were based in Dublin (including 4 Romanian developers). Besides managing specific projects, the two managers were actively involved in acquiring new projects internationally. Being an Irish-based company makes

them attractive on the international arena. In doing business, Irish companies have the reputation of being stable and reliable. The fact that they have their development division in Romania is a signal for customers that the company can offer quality work at a lower price than other competitors.

4. Methodology

The two case studies we present in this paper were researched following similar approaches, relying on qualitative ethnographic methods and an interpretivist paradigm.

4.1 Case study A

The first case study (company A) started in 2006 and has been conducted in several phases. The contact with the company was initiated during a first phase when interviews with thirteen managers and developers of German SMEs as well as four interviews with Eastern-European offshoring vendors were held. The interviews were used for identifying the challenges of offshoring for German SMEs, as well as some general strategies used to deal with them. From this sample, two companies were chosen for further analysis, one of them being Company A whose case will be discussed in this paper.

The second phase of data collection was performed using ethnographic research methods, comprising interviews, on-site observation and artifact analysis. The on-site observation involved visiting the company for two and a half weeks at the headquarters of the company in Germany. In addition, the Russian offshore partner was visited for one week. The analysis of data was based on Grounded Theory [28]. After each step, the transcripts of the material were scrutinized and coded. At first, we composed categories based on the findings in the collected data. Then, these categories were related to each other and evolved through further research.

After some of the results had been published [29, 30] research continued in Company A. This third phase involves aspects of an action research approach and aims at improving the knowledge exchange between the distributed teams by supporting articulation work [31]. Building on our knowledge of the observed work practices, we are currently conducting semi-structured interviews with all involved actors from both teams in order to refine our results and categories from the second phase.

4.2 Case Study B

The second case study (company B) was based on the findings of an exploratory study conducted in 2006 [32]. The 2006 study surveyed six small Romanian software development companies and three freelancers who were involved in outsourcing relationships as vendors. The study was motivated by the scarcity of studies exploring the challenges of outsourcing to Eastern Europe from a vendor perspective. A number of categories were identified after data coding, and these categories guided our next study. One of the conclusions regarding methods was that an outsourcing relationship needs to be studied from both ends, in order to get a more objective picture. Findings also illustrated the crucial role of cultural mediators—people who have lived in both cultures and can help each partner understand the other party's perspective.

A new study was conducted in 2007, after identifying an Irish company with a development unit in Romania. This case study focused on the challenges encountered by SMEs involved in outsourcing, with an emphasis on the role of cultural mediators in distributed software development [33]. The methods employed were ethnographically-informed: visits and observation in both sites, interviews with the two managers and two Romanian developers, collection and analysis of a number of artifacts. A new round of interviews were conducted in January 2009 for an update on the company's situation and practices as well as for validating the analysis and interpretation of the 2007 data from a new angle.

5. Findings

In this section, we will present our findings from the two cases investigated by presenting a number of work practices used to facilitate knowledge exchange. Although these practices are not novel, what we try to emphasize here are the challenges raised by both site distribution and by having to work across cultures when it comes to knowledge management.

5.1 Sharing artifacts and repositories, complemented by direct and mediated communication

In a relationship with an offshore subsidiary, many different types of knowledge need to be shared. Technical Software Engineering knowledge and domain knowledge regarding the customers' needs are paramount, but business and cross-cultural communication norms also play an important role.

One of the most important aspects of knowledge exchange between the teams is related to specification documents containing lists of features which are to be developed. In case of Company A, specifications are usually handled in the form of Microsoft Word files, which are based on the contract with the customer. Before they are sent to the remote site, the assignments in the contract are translated into English and annotated by the German project manager, who adds details concerning technical particularities of the project. However, these documents are only used as rough project guidelines during the later development. In Company B, after reaching an initial agreement, the project manager and a developer discuss the requirements with the customer (usually via call conferences) and write brief specifications to be attached to the contract.

The daily exchange of technical knowledge is more or less unstructured, highly situated and bound to emerging work trajectories, for example when unexpected problems occur, or if changes in one part of the codebase affects other modules. The communication between sites takes place mainly via *email* and *instant messaging*, which play a very important role for quick requests, for example concerning technical details of running projects. Phone and VoIP are also used frequently in the case of Company B.

Other important instruments for sharing knowledge are tools like *source code management systems* (for Company A), *live websites* using private IPs and VPNs (for Company B) and *defect tracking systems*. These are shared between sites and while their main purpose is to support the development of applications, they are also used for frequent updates on project status by project managers.

Generally, there is very little documentation available about the technologies used in the company. As a Russian developer of A explained: "(...) some specifications on features exist in the documentation (...). But documentation—for obvious reasons—never goes into details on how things are implemented. Internal architecture is not documented yet (...).“ Keeping documentation to a minimum is also one of the strategies of Company B, and like in Company A there is a strong reliance on informal communication and direct requests in case of problems.

Furthermore, the information is fragmented and can be hard to find: "(...) one notices again and again that information is there, but is distributed in a way that makes gathering it cumbersome..." (Developer, Company A). Hence, in many circumstances, rather than looking for information stored in various

databases, emails or chat-logs, people prefer to simply ask local or remote colleagues. This practice can lead to problems for the cooperation, when one team needs information from the remote site and does not get an answer, which is not unusual according to several developers in company A.

The current practices in dealing with documentation and artifacts in the two companies can be interpreted as an organizational culture issue, SMEs being known for often adopting agile development methods. However, there seem to be also differences between the onshore and offshore teams, as German project managers and developers reported. According to that, the Russian developers simply do not like to write documentation. Instead, they would prefer to write code which is “self explanatory”, and not linger with documentation which—according to them—would be outdated most of the time anyway. Hence, according to the German side, when the Russians were requested to send documentation on one particular feature, they would write it down on demand. This focus on programming as opposed to other aspects of software engineering work is accompanied by the temptation to redesign existing technical frameworks instead of focusing on the requested features. As a German project manager put it: “All developers are architects-to-be, too. (...) You want to have a car door repainted, and get a new vehicle.”

Interestingly, in our interviews with the Russian developers, there was a different view on the role of documentation. From the perspective of the Russian team leader, the Russians wrote much more documentation than the Germans, who often ignored these tasks. Sometimes this lead to problems, like in the case when a German project manager had simply forgotten to update the specifications with some change requests from the customer, and the Russian team worked several weeks on features which had been dropped. Although this is an extreme example, it illustrates how different organizational practices look from each perspective.

5.2 Meeting as a way of keeping up to date

Company A has regular weekly meetings at its German headquarters to give people an overview on what is going on in the company, discuss current developments and problems and share information on new technologies and tools that may be useful for the team as a whole. The offshore team in Tomsk is holding a similar meeting. Both teams write minutes which are meant to summarize the discussions, and they exchange them with each other.

However, as both developers and project managers reported, information shared during the meetings as well as by exchanging the minutes is not very useful for keeping up-to-date: “(...) if all I know (...) is that a developer has worked on this or that... this is somehow sparse information“. Hence, the developers and project managers explained they would rather keep aware of what was going on by going around and talking to people.

Starting with 2008, the Russian team members also have to write brief minutes of their weekly meetings and send them to the German team. These minutes are valued by some German developers, as one of them explained: “(...) it is like with any weekly meeting minutes, the information is in many parts meager (...). But I like to read them, sometimes I can find something new, unknown, or I realize, ‘ah, they are working on the same problem I worked on some time ago!’, or if I am waiting for a solution (that is developed at the remote site), and so on“. In this regard, the short references to what is going on in Tomsk are used as props for direct requests and communication by dedicated German developers, but the minutes are not necessarily used as a medium for exchanging knowledge directly.

In contrast with these separate weekly meetings and exchange of minutes between the groups, in Company B the two managers meet every morning on *Skype* and review the status of each project. They coordinate their activities for the day and divide the tasks. During the workday, they permanently maintain an open communication channel not only between themselves, but also with the developers. This practice is probably a result of their long collaboration. Mirroring this practice, the Romanian developers working jointly with customer development teams also maintain open channels with their counterparts throughout the day. Managers also have almost daily conversations with each developer, usually via instant messenger, to get updates on the status of specific tasks.

As in the case of any small company, awareness on what the other colleagues are doing is also maintained in less formal ways, during smoking breaks (Romanian developers), over lunch (Russian developers) or simply by going around the company and talking to colleagues (Russian developer living in Germany). This practice seems to be very important in case of company A. For example, one of the German developers also came across as an informal knowledge broker: „(...) often I can give information on things which are actually not my responsibility but I happen to have heard from another—ok, this works this way, this was built that way“. He continued: “(...) I walk around and simply ask

‘what are you working on?’ And they say ‘I am developing an application, I have to take screenshots’. Then I might say: ‘(...) try to do it with this (tool), it could make your life easier, or maybe not ... (laughter)’. However, this practice is only possible at the local site, while information on the remote team is usually sparse and obtained by referring to minutes, initiating chat communications, or personal visits.

5.3 Cross-Cultural Perceptions and Mediation

People from different cultures have different backgrounds and they encode and decode messages differently. If the partners ignore this reality, they tend to assume that everyone's thoughts and actions are just like theirs—and this increases the chances of misunderstandings.

In regard to general cultural issues, we heard different perceptions during the interviews. Hence, the German manager of Company A explained that the Russian developers would have “a high motivation working on specific tasks. But if a job gets monotone, the performance decreases quickly.” At the same time, in his view, the Russians seem to have a „higher endurance compared to the local personnel“, thus being able to keep their performance high under pressure over longer periods of time. The Irish manager of Company B saw Romanian developers as having a “great desire to be successful” and praised their dedication, showing that they occasionally worked late at night and during week-ends to get the job done or solve a problem.

However, in Company A, the assistant manager also spoke about what she perceived as an exaggerated sensitivity to criticism on the part of the Russian developers. From her perspective, German developers are less inclined to take criticism personally and are more emotionally detached from their work. At the same time, a Russian developer (who was living in Germany at that time for a year) complained about what he called “an (organizational) culture of blaming each other” in the company. He complained that most of the communication of his German colleagues with the Russians would consist of criticism, and positive developments are not acknowledged properly.

On a totally different note, a Romanian developer mentioned about his communication with the Irish customer team: “their emails are always so nice, and they all end on an optimistic tone; problems are signaled in such a polite note, that you have to read an email several times to understand there is a problem”.

Managers in both companies acknowledged and praised the high level of technical skills of their

offshore developers, but were not satisfied with their business communication skills. While domain knowledge related to project work was relatively easily appropriated by Russian and Romanian developers, acquiring business communication skills proved to be a long term process and created problems in the case of Company B, where developers were given the responsibility of managing the relationship with the customer. Similarly, there were problems in company A as the Russian team set up an own homepage, which included confidential technical information (for example IP addresses of internal services like the CVS) as well as unlicensed copyrighted images. As the German side demanded taking down the homepage, the Russian team apparently had no understanding for the legal problems their action could entail. Instead, they wrote back asking: “Why are you starting war on this?” According to the Russian developer staying in Germany, this had to do with the prevailing “culture” of blame, which affected the interpretations of the event in a negative way.

In both companies there are people bridging the two cultures who have also notable technical and domain knowledge. They are acting naturally for managing and mediating communication, work with both sides and spend time working on each site.

In Company A, the Russian developers living in Germany act as mediators between the sites. “I am frequently getting requests from (the German manager) or from (the Russian team manager) to improve communication. So, then what am I doing? I am running around, asking people what is the status of different things, what are the difficulties in communication, what are the points where people feel dissatisfied with the other party’s work. And then I try to create a kind of neutral technical description of the situation. It worked so far”. In regard to his role in the company, he further explained: “I think I became part of the German team—for sure, because my normal working routine involves working here with the German team. I have a cultural connection and some mental connection with the Russian team, of course. It saves a lot of time, effort and emotions that I understand the language, that I can hear their complaints (laughter)“.

In company B, the Romanian manager plays a paramount role in running the company; her 7 years spent in Ireland working closely with her Irish counterpart gave her the chance to acquire valuable domain knowledge and business skills, and also have been the basis of the shared understanding they developed. Whether spending time on the Romanian site or traveling to acquire new customers, she has

permanent open channels with the other manager and with the Romanian developers. During the interview, the Irish manager spoke about how collaboration with Romania would have been a totally alien idea to him ten years ago, but having the Romanian manager on his team for four years before starting the current company has given him confidence in her skills and consequently in the people she recruited in Romania.

Interesting enough, the Romanian manager said that if she would decide to live in Ireland for a longer period, she would hire a Romanian and not an Irish manager to run the company in Bucharest. At the same time, in Company A, there are preparations under way to send a German project manager to run the Russian site.

5.4. Spending time on the other site

Interviews in both companies have revealed that personal *face-to-face* contact plays a very important role in knowledge exchange. Besides building trust in the skills of remote team members, personal meetings have an important role in learning how to approach a person from the other site. The face-to-face meetings constitute an important basis for building social ties, reinforced by exchanging informally personal information online (about family events, kids going to college, health issues). The existence of social ties has been shown to improve knowledge transfer and communication in general.

In order to deal with the still prevailing communication issues between the teams, Company A is supporting regular visits of their staff to the Russian site and also tries to invite Russian developers over to Germany for longer periods of time. The motivation of this practice is threefold: first, the company wants to support mutual enculturation. Second, the stays are meant to support the knowledge exchange between the sites. Being an expert programmer, the Russian developer is asked to share his knowledge with the German colleagues, and serve as contact person for the Germans. And, last but not least, the opportunity to live in Germany for a period of time is also meant as a motivation for the Russian developers to continue working with the company. As the German assistant manager explained: "If new employees are hired, then this is an incentive for them to accept the position, because then they are invited to come to Bonn for three months and spend time in Germany".

Company B also facilitates brief visits of Romanian developers to customer sites, perceived as direct contact opportunities and marking important phases in the project. During these visits, developers get the

chance to gain a better understanding of the environment their counterparts are working in, to see them at work and learn from their practice. While this is fairly easy to organize for European customers, visits to the US are more complicated to organize.

Generally, these mutual visits are highly appreciated by the developers on both sides. One of the German developers of A explained: "I myself have realized that the contact became much better after (some of the Russian developers) have been on site, I would say. Often, especially in regard to technical details or to the design of a user interface, the communication over the Internet was rather slow. And then, when we sat together face to face, and I made a few gestures, and showed what I wanted, the understanding came much quicker (...)". Furthermore, the visits endorse personal contacts between the teams, as formal work visits are usually accompanied by private activities. For example, one of the German project managers reported he would like to spend a weekend skiing with the Russian developers when he would be in Tomsk. Furthermore, during the on-site observation in Tomsk, the Germans were invited to a bowling center during the lunch break by a Russian developer celebrating his birthday. These events play a paramount role for socializing, as both teams like to show their guests around during their stays. Some of the Russian and German developers also reported, that these relationships had developed into personal friendships during several visits.

Recently, some of the best Romanian developers were invited to spend a longer period of time in Ireland working with customers there. Contrary to the expectations of the Irish manager, while any young Irish developer would have been thrilled by such an opportunity, Romanian developers were not very enthusiastic about spending extended periods of time abroad and this resistance surprised him.

6. Discussion

By investigating actual work practices in their social (and cultural) embedding in two small companies, we aimed at gaining a new understanding of cultural influences on knowledge management in global teams as recommended by Granovetter [16].

An important factor for knowledge management we found in the field was related to the concept of social capital and 'bridges' between the teams [14]. In this regard, company A is an example of how social capital can hinder knowledge exchange, as well as stimulate it [17]. The conflicts between the teams in regard to dealing with documentation, the "culture" of blaming

each other as well as different perceptions of critique indicate a lack of shared social capital (which would help to resolve conflict situations). It became apparent that both companies rely heavily on some key people, who act naturally as information brokers and conflict mediators. In this regard, the knowledge exchange between the teams was heavily reliant on relationships between particular developers who held a high social capital in both teams. Company B, on the other hand, had far less problems as the communication between the sites was “channeled” through the two managers who shared a high level of social capital, each with his own team, as well as with each other.

Practices of building social capital were closely related to visits on the other site, but also to private initiatives of developers who befriended colleagues from the other team. These personal ties helped bridging the distance and resolving problems which were partially related to cultural differences from the perspective of our interviewees.

Our findings offer an alternative to the discourse about “culture” as a distinct factor that puts an emphasis on the differences between national cultures [20]. In our view, cultural factors are rather intertwined with other complex issues—like social ties, informal communication, as well as micro-politics—that have to be studied in context. An interesting aspect is that all the collaborations presented here could be considered pan-European (Tomsk lies in western Siberia at the border to Asia), and the general view assumes that cultural differences between European countries would be minimal. Although this is true to a point, our studies showed that there is a notable difference in the way offshore members of the team operate in organizations (especially at the business communication level).

The strategy the two companies used for managing knowledge corresponded to what Huysman and De Wit [12] called “the second wave of knowledge management”: most of the knowledge sharing activities were informal, and the tools used were deeply grounded in the organizations’ practices. In this regard, they showed certain resemblances to coordinative practices as implied by the concept of articulation work [30, 31].

Regarding work practices involved in knowledge transfer, the examples show that a lot of interaction happens in an informal, improvised way. The procedures in place are more based on people and direct or mediated communication, and less on tools, repositories and artifacts. In small companies, knowing in practice (knowing who is working on what, who knows what, who is working with whom) seems to be far more important than following a standardized

procedure and following every prescribed step. In this respect, our findings concur with those of Pyoria, that “knowledge intensive organizations should always value human relations above technology” [34].

7. Conclusion

The cases presented in this paper are by no means representative, and our aim was rather the illustration of the intricacies of knowledge work in a distributed software development setting. In our opinion, there is a need for further research in this area from a combined social and technical perspective.

There are already approaches to support second wave knowledge management by groupware tools which aim at improving informal KM practices of particular communities [13, 35]. In order to be successful, these tools need to be grounded in the work practices of the communities they aim to support.

It is not yet clear how these tools can be adapted to the distinct practices of global software engineering teams. In this respect, more in situ studies of real organizations would have the potential to enhance our understanding of existing challenges and of the generalizable aspects of solutions that proved successful in practice.

8. Acknowledgments

The studies reported in this paper were partially supported by: The German Research Foundation (DFG) within the research project ARTOS, and Lero (the Irish Software Engineering Research Centre) within the socGSD cluster project funded by the Science Foundation of Ireland (SFI).

9. References

- [1] F.O. Bjørnson and T. Dingsøy, „Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used,” *Information and Software Technology*, vol. 50, Okt. 2008, pp. 1055-1068.
- [2] H. Timonen and M. Jalonon, „A Critical Review of Knowledge Management Literature: Introducing a Practice-based Approach on Knowledge Sharing ,” 9th European Conference on Knowledge Management, Southampton, UK: 2008
- [3] M.S. Ackerman, V. Pipek, and V. Wulf, *Sharing expertise: beyond knowledge management*, Cambridge, Mass.: MIT Press, 2003.
- [4] Y. Hsieh, “Culture and Shared Understanding in Distributed Requirements Engineering.”, In *Proceedings of the IEEE international Conference on Global Software*

- Engineering (October 16 - 19, 2006). ICGSE. IEEE Computer Society, Washington, DC, 101-108.
- [5] K. V. Siakas, E. Georgiadou, "Empirical Measurement of the Effects of Cultural Diversity on Software Quality Management." *Software Quality Control*, Vol. 10, 2 (Sep. 2002), pp.169-180.
- [6] G.H. Hofstede, *Culture's consequences: comparing values, behaviors, institutions, and organizations across nations*, Thousand Oaks, Calif.: Sage Publications, 2001.
- [7] T.H. Davenport, L. Prusak, *Working Knowledge: How Organizations Manage What They Know.*: Harvard Business School Press, Boston, MA, USA, 1998.
- [8] Manifesto for Agile Software Development, <http://agilemanifesto.org/>, Access: 20.01.2009.
- [9] A.F. Buono, F. Poulfelt, *Challenges and Issues in Knowledge Management.*: Information Age Publishing, Greenwich, CT, USA, 2005.
- [10] L.A. Suchman, *Plans and situated actions: the problem of human-machine communication*, Cambridge, New York, Port Chester, Melbourne, Sidney: Cambridge University Press, 1987.
- [11] M. Polanyi, *The tacit dimension*, London,: Routledge & K. Paul, 1967.
- [12] M. Huysman, M. and D. de Wit, "Practise of Managing Knowledge Sharing: Towards a Second Wave of Knowledge Management," *Knowledge and Process Management*, vol. 11, No 2, 2004, pp. 81-92.
- [13] T. Reichling, M. Veith, and V. Wulf, "Expert Recommender: Designing for a Network Organization," *Computer Supported Cooperative Work (CSCW)*, vol. 16, Okt. 2007, pp. 431-465.
- [14] A.E. Milewski, M. Tremaine, R. Egan, S. Zhang, F. Kobler, und P. O'Sullivan, "Guidelines for Effective Bridging in Global Software Engineering," *International Conference on Global Software Engineering*, Los Alamitos, CA, USA: 2008, pp. 23-32.
- [15] J.D. Herbsleb, and R.E. Grinter, "Architectures, coordination, and distance: Conway's law and beyond" *Software*, IEEE , vol.16, no.5, pp.63-70, Sep/Oct 1999
- [16] M. Granovetter, „Economic Action and Social Structure: The Problem of Embeddedness," *American Journal of Sociology*, vol. 91, 1985, pp. 481-510.
- [17] M. Huysman und V. Wulf, *Social capital and information technology*, Cambridge, Mass.: MIT Press, 2004.
- [18] D. Cohen und L. Prusak, *In good company: how social capital makes organizations work*, Boston: Harvard Business School Press, 2001.
- [19] P. Bourdieu, "The form of capital," *Handbook of theory and research for the sociology of education*, J.G. Richardson, ed., New York: Greenwood Press, 1986.
- [20] M. Gallivan und M. Srite, „Information technology and culture: Identifying fragmentary and holistic perspectives of culture," *Information and Organization*, vol. 15, 2005, pp. 295-338.
- [22] E. MacGregor, Y. Hsieh, und P. Kruchten, „Cultural Patterns in Software Process Mishaps: Incidents in Global Projects," *International Conference on Software Engineering*, St. Louis: 2005, pp. 1-5.
- [23] J. Winkler, J. Dibbern, und A. Heinzl, „The impact of cultural differences in offshore outsourcing—Case study results from German–Indian application development projects," *Information Systems Frontiers*, vol. 10, Apr. 2008, pp. 243-258.
- [24] B. McSweeney, 'Hofstede's Model of National Cultural Differences and Their Consequences: A Triumph of Faith – A Failure of Analysis'. *Human Relations*, Vol. 55/1, 2002, pp: 89-118.
- [25] B. Anderson, *Imagined communities. Reflections on the origin and spread of nationalism*, London [u.a.]: Verso, 2006.
- [26] C. Geertz. „Thick Description: Toward an Interpretive Theory of Culture“. In *The Interpretation of Cultures: Selected Essays*. New York: Basic Books, 1973, pp. 3-30.
- [27] Worldmap by I.K. Sankakukei, <http://english.freemap.jp>, distributed under a CC-BY 3.0 license.
- [28] A.L. Strauss und J.M. Corbin, *Basics of qualitative research: techniques and procedures for developing grounded theory*, Thousand Oaks: Sage Publications, 1998.
- [29] A. Boden, B. Nett, und V. Wulf, „Coordination Practices in Distributed Software Development of Small Enterprises," *International Conference on Global Software Engineering*, 2007, pp. 235-246.
- [30] A. Boden, B. Nett, und V. Wulf, „Articulation work in small-scale offshore software development projects," *Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering*, Leipzig, Germany: ACM, 2008, pp. 21-24.
- [31] A.L. Strauss, "The Articulation of Project Work: An Organizational Process," *The Sociological Quarterly*, vol. 29, 1988, pp. 163-178.
- [32] G. Avram, "Developing Outsourcing Relationships: A Romanian Service Provider Perspective", *The First Information Systems Workshop on Global Sourcing – Services, Knowledge and Innovation*, Val d'Isere, France, 13-15 March 2007.
- [33] I. Richardson, G. Avram, S. Deshpande, and V. Casey, "Having a Foot on Each Shore - Bridging Global Software Development in the Case of SMEs." In *Proceedings of the 2008 IEEE international Conference on Global Software Engineering - Volume 00 (August 17 - 20, 2008)*. ICGSE 2008, IEEE Computer Society, Washington, DC, 13-22.
- [34] P. Pyoria, „Informal organizational culture: the foundation of knowledge workers' performance" *JKM*, 11/3 2007, pp.16-30
- [35] V. Pipek und V. Wulf, "Pruning the answer garden: knowledge sharing in maintenance engineering," *Proceedings of the eighth conference on European Conference on Computer Supported Cooperative Work*, Helsinki, Finland: Kluwer Academic Publishers, 2003, pp. 1-20.