

## Knowledge Sharing Practices and the Impact of Cultural Factors: Reflections on Two Case Studies of Offshoring in SME

Alexander Boden	Gabriela Avram	Liam Bannon	Volker Wulf
University of Siegen	University of Limerick	University of Limerick	University of Siegen
<a href="mailto:alexander.boden@uni-siegen.de">alexander.boden@uni-siegen.de</a>	<a href="mailto:gabriela.avram@ul.ie">gabriela.avram@ul.ie</a>	<a href="mailto:liam.bannon@ul.ie">liam.bannon@ul.ie</a>	<a href="mailto:volker.wulf@uni-siegen.de">volker.wulf@uni-siegen.de</a>

### Abstract

*The impact of culture on knowledge management in international teams is an important topic which is still not well understood. We want to contribute to the discussion by presenting two case studies of small software teams involved in distributed software development. In doing so, we illustrate how cultural and social issues influence the way knowledge exchange is performed by analyzing four knowledge sharing practices: status meetings and maintaining awareness, the collaborative use shared artifacts and repositories, spending time at the other site and human “bridges” that mediate between people and cultures. Our findings suggest that organizational culture is permanently re-negotiated and adjusted to fit the distributed collaboration, as the teams learn how to deal with each other. Socialization plays a significant role in this learning process, and people are more likely to draw on national stereotypes when breakdowns occur. The influences of national culture and site-specific organizational culture are subtle and not easy to separate from other factors. Based on our experience, we argue that in order to achieve an accurate understanding of knowledge sharing practices in globally distributed software teams, these need to be studied in context, longitudinally, and from both onshore and offshore perspective.*

**Keywords:** Global Software Engineering, Culture, Small Enterprises, Knowledge sharing practices, Offshoring, Ethnographically-informed studies

### 1. Introduction

The ongoing economic incentives and pressures of globalization have led to the growth of distributed international organizations that attempt to take advantage of time differences and variations in labor availability and cost in order to improve competitiveness. The software industry is one of a number of fields which has been influenced by this increase in globalization, leading to a variety of business strategies that attempt to harness the possibilities of, for example, offshoring certain activities to lower-cost sites, developing globally-distributed software teams that can work on projects 24/7, etc. As software and source code can be transferred easily between globally distributed sites, offshoring of software development has been widely seen as a means for cost reduction and efficiency gains. However, 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. A growing literature base exists on the delineation of these problems, and attempts to resolve them, in the context of the emerging sub-field of software engineering known as Global Software Engineering (GSE). Problems of communication, coordination, and collaboration in the software domain are also studied in such fields as Information Systems (IS), Computer Supported Cooperative Work (CSCW), and Knowledge Management (KM). However, there are still many unresolved issues concerning the management of such globally-distributed development efforts.

In this paper, we provide a contribution to this on-going research field through documenting and analyzing some of the knowledge-sharing practices and communications of distributed international teams of software developers. We make a number of observations concerning the problems encountered. Our approach is one that focuses on the work practices of the different software teams, and the many and varied ways in which they manage to accomplish their work, despite the difficulties of communicating across languages, cultures, time and distance. Along the way, we critique certain GSE approaches to the study of these distributed groups that overly rely on what we believe to be limited conceptions of such key concepts as culture and knowledge.

With regard to knowledge, there has been a focus on knowledge management (KM) approaches for understanding the organization and behavior of distributed teams by referring to “canonical” concepts of knowledge as a product—suggesting that most 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 focus on understanding how knowledge is embedded in social relationships and how actors actually share and put their knowledge into practice [3,4].

One important issue with international teams is the impact of culture—under multiple aspects: national, organizational, and professional. While the topic of “culture” is one that has captured the interest of the SE community for some time [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 knowledge sharing work practices in GSE. We want to contribute to the discussion by presenting two case studies of small-size software teams involved in distributed 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 enterprises.

The paper is organized as follows: after a discussion of the related literature (section 2) we introduce 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 (section 7).

## **2. Related Work**

In our study, we focus on work practices—with an emphasis on knowledge sharing activities—and we attempt to provide a better understanding of the impact of social and cultural factors on these activities. As a basis for our discussion, the following sections are meant to provide an overview on our perspective of these concepts.

## **2.1. Knowledge in (Global) Software Engineering**

The research literature offers different conceptualizations of the role knowledge plays for software engineering. While studies in the field of GSE are adopting both *technocratic* and *behavioral* approaches, technocratic approaches are clearly dominating the scene [1]. This focus on rather traditional knowledge management concepts is problematic, as it supports a view that considers knowledge as being an object that can be de-contextualized, captured, and disseminated “on demand”, without any loss of meaning, through information systems.

In contrast, behavioral approaches stress that although information can be represented in the form of explicit content, it needs to be contextualized in order to become knowledge again. Furthermore, knowledge needs to be framed in order to contribute to the expertise needed in practice [7]. Knowledge sharing, therefore, should not be considered as mere consumption of information, but as a complex and reflexive practice of cooperating actors [3]. This is reflected in a broad set of theories claiming that action is situated [7] and deeply connected to tacit knowledge [8], which cannot be made entirely explicit. Huysman and de Wit have labeled this shift of focus toward tacit and emergent aspects of knowledge as the ‘second wave’ of KM [8]. In this socio-technical understanding of KM, the focus moved from setting up canonic knowledge databases to supporting informal knowledge sharing of communities with tools grounded in the practices of the particular fields [9]. Hence, knowledge is rather thought of as being socially embedded, and appropriate strategies take into consideration the practice-related aspects of knowledge management. Orlikowski [10] has hinted at knowing-in-practice as an important element for organizational operation by illustrating how knowledge was enacted and (re-) constituted through several practices in a distributed organization (such as sharing identity, interacting face-to-face, aligning efforts, learning by doing and supporting participation).

While these analytical concepts are concentrated on the various ways of handling knowledge in practice, other perspectives focus on the social structure supporting knowledge sharing. These approaches are interested in how social relationships shape the way knowledge is shared in practice. For example, Granovetter [11] has emphasized the role of social connections for the functioning of organizations. Also Marczak et al. [12] have shown the importance of social networks in fostering relationships, trust and knowledge management. Building on these concepts we want to analyze the role of culture for knowledge management practices in international teams.

## **2.2 Cross-cultural Aspects of Global Software Engineering**

Cultural compatibility is often described as an important factor (amongst others) in determining the success of collaboration in international software development teams [13]. The impact of culture on software development—be it national, organizational, or professional, culture—is a topic with a 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 [14,15]—and also to a discussion of the impact of culture on knowledge management practices in GSE [16].

The cultural terms used in the GSE literature often focus on national aspects of intercultural work [17,18]. These approaches usually treat culture as being equivalent to national identity,

referring to Hofstede's framework of cultural dimensions [6]. Thus, one sees survey studies done on the differences in communication style between North American engineers and Indian engineers for example, which seem to assume that one can work with such generic categories based on geographic location or national identity. Usually, such studies attempt to compare national cultures by operationalizing variables like "power distance", "individualism", or "uncertainty avoidance" which are expected to represent characteristic attitudes shared amongst the citizens of a nation (and/or members of an organization). Within the organizational studies field, Hofstede's formulations have been the subject of extensive conceptual and methodological critique (see, for example [19]). 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; and 3) actors are allocated to one culture at a time, while different cultures are seen as being mutually exclusive. Without wishing to be drawn into this controversy here, we do urge caution in the use of such self-report survey instruments to investigate globally-distributed software teams. 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".

Other researchers have developed 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 [4,26,29]. In accord with such interpretivist approaches, we propose a much broader understanding of culture: we see culture as a shared web of meanings that shapes roles and interpretations, and is dynamically (re)negotiated by the actors in the course of their daily work. Hence, we are more interested in the actors' interpretations and related processes of sense making, than in the definition of cultural particularities [20].

This broad understanding of culture entails many different layers referring to national, professional, or religious aspects, which are seen as being intertwined in a complex, non-hierarchical way, and which can hardly be studied in isolation [14]. 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 the form of artifacts, practices, and routines (as well as the related interpretations and connotations), which will be in the center of our attention in this paper.

In the following, we adopt this more nuanced interpretive approach to examine two case studies of globally-distributed software development. Our aim is to look at what can we learn from practice, with a view to developing our understanding of what kinds of software tools and organizational support mechanisms might be required.

### **3. Cases**

In this section we will provide a short overview on the two case studies we have conducted (see Figure 1).

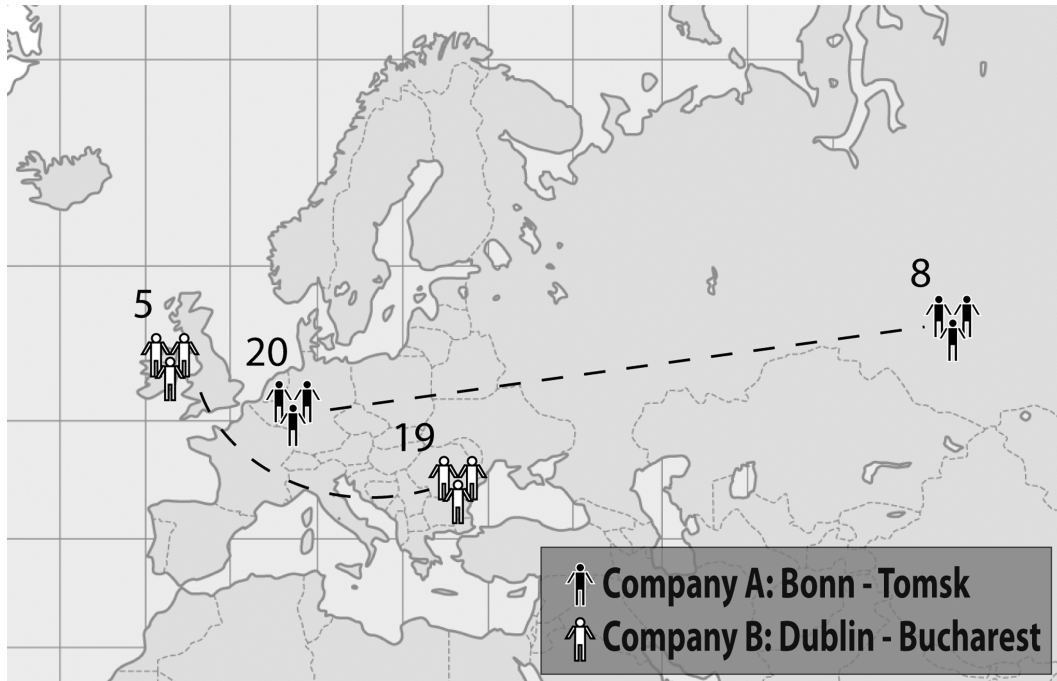


Figure 1. The locations of the teams. Numbers indicate the team sizes.  
(worldmap by I.K. Sankakukei, <http://english.freemap.jp>, distributed under a CC-BY 3.0 license)

### 3.1. Company A Overview: Germany (Bonn) – Russia (Tomsk)

**Company A** is a small German software enterprise engaged in the field of statistics and documentation located in Bonn. The customers are mainly German archives and museums. The company was established in 1980 and had approximately 20 employees in 2009.

In the mid-1990s, the company found it increasingly difficult to hire German developers, as wages had increased considerably and the labor market had 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, a number of four to eight employees are working for Company A in Tomsk. 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 shifting toward 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 cooperation with the offshore site was expanded to several small size projects, using Russian developers—working in close cooperation with a German project manager—for customizing software products to the specific needs of particular customers. The long lasting cooperation (more than 10 years) made Company A an interesting case for our study.

### 3.2. Company B Overview: Ireland (Dublin) - Romania (Bucharest)

**Company B** was established in January 2006 in Dublin, Ireland. The two owners had worked together in a company providing software applications for telecoms and media companies for four years. During that period, one of them had been a project manager and the other (originally from Romania) had been working on his team as a senior developer. In January 2006, following changes in the management of their employer, they decided to leave their employment and set up their own company. They hired four 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 more developers in Ireland, but failed due to the harsh competition. Therefore, the Romanian project manager identified a small company with five employees in Bucharest, Romania, which they consequently 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 founder) in Dublin, with the Romanian manager 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 are 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, and the fact that they have their development division in Romania is a signal for potential 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 investigated using similar approaches, relying on ethnographically-informed methods and based on an interpretivist paradigm.

##### **4.1 Case study A: Research Methods**

The first case study (Company A) started in 2006 and has been conducted in several phases. It aimed at understanding coordination and organizational learning in small-sized, distributed software teams [21,22]. 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 people from Eastern-European offshore companies were held. The interviews lasted roughly one hour each and were used for identifying the challenges of offshoring for German SMEs, as well as some general strategies used by these companies to deal with them.

The second phase of data collection was performed using ethnographic research methods, comprising interviews, on-site observation and artifact analysis at Company A. The on-site observation involved spending two and a half weeks at the company's headquarters in Germany. In addition, we spent one week on the site of the Russian offshore partner. We also conducted thirteen extensive interviews with members of the German and Russian teams, including developers, as well as people from the management. The interviews lasted between one and two hours each and were used to analyse the different perspectives on the work practices we had identified during our observations. In order to check our findings, we discussed the results of our research with the practitioners during a workshop.

## **4.2 Case Study B: Research Methods**

The second case study (Company B) was based on the findings of an exploratory study conducted in 2006 [23]. The 2006 study surveyed six small Romanian software development companies and three freelancers who were involved in outsourcing relationships as vendors. A number of categories were identified after data coding, and these categories guided our next study. One of the conclusions regarding methodology was that an outsourcing relationship needed to be studied from both ends, in order to get a more objective picture.

A new study was conducted in 2007, after identifying an Irish company with a development unit in Romania- Company B in the current study. 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 [24]. The methods employed were ethnographically-informed: two visits and observation on both the onshore and the offshore site, four interviews (with each one of the two managers and two of the Romanian developers), examination of artifacts. A new round of interviews was conducted in January 2009, for getting an update on the company's situation and practices, as well as for validating the analysis and interpretation of the data collected in 2007 from a new angle.

## **4.3 Data Analysis for the Current Study**

The frame for this study was a result of a discussion between the authors on their ongoing research at the time, revealing similarities and differences between the two cases. The data was then analyzed jointly, by taking different aspects of knowledge management from the literature and discussing their manifestation in each of the cases. Based on the discussions, we identified several practices of knowledge management which were relevant to our cases. They will be presented in the following section, followed by a discussion of cultural aspects entwined with knowledge management practices in these distributed software teams. For describing each practice, we relied on our own observations, on what our informants said during the interviews and on the analysis of artifacts that were made available to us (meeting minutes, chat archives, email exchanges, requirements documentation).

## **5. Research Findings**

In this section, we will connect our findings from the two cases, presenting a number of work practices used by the two companies for facilitating knowledge exchange. In line with our own understanding of culture, we will focus on three relevant aspects:

- Practices: How do the practitioners share knowledge?
- Interpretations: What is their own perspective on how they are doing it?
- Artifacts: What is the role of related documents, tools and media in this regard?

Our analysis of the settings is organized around a number of topics which have been shown to be of crucial importance in helping the distributed teams in our two cases to coordinate their work and ensure that the work was accomplished. The challenges introduced by having to work across cultures are also addressed.

## 5.1 Status Meetings and Maintaining Awareness

An important practice we encountered in both cases was that of regular meetings (co-located or distributed) and status information exchanges between sites.

Company A had regular weekly meetings at its German headquarters, in order to give people an overview on what was going on in the company, discuss current developments and problems and share information on new technologies and tools that might be useful for the team as a whole. The offshore team in Tomsk was holding a similar weekly meeting. Starting with 2008, the team members in Tomsk were required to write brief minutes of their weekly meetings (in English) and share them with the German team. As both developers and project managers reported, information shared during the meetings (and in form of the minutes) was not sufficient 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 that they would rather stay aware of what was going on by going around and talking to people—a practice mainly possible in collocated environments. Furthermore, the short references to what was going on in Tomsk were sometimes used as props for direct requests and communication by some of German developers, but the minutes were not considered as a medium for exchanging knowledge directly.

Company B had a different approach: the two managers started every day with a status review of all their ongoing projects over Skype. They coordinated their activities for the day and divided the tasks. During the workday, *they permanently maintained an open communication channel*. This practice was probably a result of their long-term collaboration. Mirroring this practice, the Romanian developers working jointly with customer development team members also maintained open channels with their remote counterparts throughout the day. It was also customary for the managers to have regular conversations via instant messenger with each developer in order to get updates on the status of specific tasks.

Status meetings are common practice in Global Software Engineering; in these two particular cases, the particular organizational cultures and power structures have lead to different practices that best suit the two companies.

## 5.2 Collaborative Use of Shared Artifacts and Repositories

For software development, be it local or distributed, knowledge related to the application domain and the aims of the development play a very important role for successful collaboration of developers. Usually, companies use artifacts like project plans and specification documents as boundary objects for supporting the necessary knowledge exchange and coordination between the customer, the manager and the (in this case distributed) developers [21].

In case of Company A, specifications were handled in the form of text files, compiled by the German project managers based on the contract with the customer. Before they were sent to the remote site, the assignments in the contract were translated into English and commented on by the German project manager. However, these documents were only used as rough project guidelines during the later development, and were not always maintained up-to-date. Generally, we found there was very little documentation available about the technologies deployed in both companies. As a Russian developer of Company A explained: “(...) some specifications of features exist in the documentation (...). But documentation—for obvious reasons—never goes



into details on how things are implemented. The 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. After reaching an initial agreement, the project manager and the developer assigned to the task discuss the requirements with the customer (usually via call conferences) and the developer is asked to write brief specifications in English (double checked afterwards by the project manager and the customer) to be attached to the contract.

In general, our informants reported that the information is fragmented across various databases, emails or chat-logs and can be hard to find: “(...) one notices again and again that information is there, but is distributed in a way that makes collating it cumbersome...” (Developer, Company A). Hence, in many circumstances, rather than looking for information in the documentation, people prefer to simply ask local or remote colleagues. The prevailing reliance on communication in the absence of documentation can lead to breakdown situations when one team needed information from the other site and could not get a prompt answer—as several interviewees from Company A told us. However, the reasons for the coordination breakdowns were assessed quite differently on behalf of the different teams:

According to German project managers and developers, the Tomsk developers simply did not like to write documentation. Instead, they preferred to write code considered by them “self explanatory”, and not linger with documentation which “would be anyway outdated most of the time”. This focus on programming as opposed to other aspects of software engineering work was 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 instead you get a new vehicle.” This situation seems to be more the result of specialization and division of tasks between project managers, architects and developers than an effect of geographical distance.

The Russian developers we interviewed had a different perspective on the documentation issue. The Tomsk team leader found that his developers wrote much more documentation than their counterparts in Bonn, who often ignored these tasks. The case of a German project manager who “forgot” to update the specifications with change requests from the customer, making the Russian team work on features that had been dropped for several weeks, was mentioned. Again, this appears to be due to the lack of coordination between the two professional groups rather than due to differences in national cultures.

### **5.3 Spending Time at the Other Site**

Interviews in both companies have revealed that personal *face-to-face* contact plays a very important role in knowledge exchange, confirming what the previous research on GSE had already shown [25]. Besides building trust and getting a realistic perspective on the skills of remote team members, personal meetings have a significant 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 that can be reinforced later by exchanging informally personal information online (about family events, kids going to college, health issues, etc.—as mentioned by the Irish manager of Company B).

In order to deal with the prevailing communication issues between the teams, Company A supported regular visits of their staff to the Tomsk site and also tried to invite Russian developers over to Germany for longer periods of time. The motivation for this practice was threefold: to support mutual enculturation, to support the knowledge exchange between sites and to serve as an incentive in preventing Russian developers from leaving the company.

In Company B, brief visits of Romanian developers to customer sites, perceived as direct contact opportunities and marking important phases in the project were systematically organized. During these visits, developers got 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. The Irish manager paid regular visits to the Romanian site, while project managers and developers from the customer side also spent time in Bucharest working with the Romanian developers. Several informants told us that once back to their desks from visiting customer sites, they were proactively sharing what they had seen and learned with their colleagues.

Generally, these mutual visits were highly appreciated by the developers on both sides. One of the German developers of Company 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 teams, as formal work visits are usually complemented by social activities. For example, one of the German project managers reported he followed the personal invite of one of the Russian developers to spend a weekend skiing when he was in Tomsk. Furthermore, during the on-site observation in Tomsk, the German guests were invited to a bowling center during the lunch break by a Russian developer celebrating his birthday. These events played a significant role in socialization, as both teams liked to show their guests around during their stays.

#### **5.4 Human “Bridges”: Mediating between People and Cultures**

In both companies, we found people bridging the two cultures who also have notable technical and domain knowledge. They were very important for managing and mediating the communication between the teams by working with both sides and keeping in constant touch with the other team.

In Company A, several Russian developers living in Germany acted 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 moving around, asking people about the status of different things, the difficulties in communication, 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 in Bonn with my German colleagues. I have a cultural connection and some psychological connections 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)”. Knowledge exchange between the teams not only required the translation of documents, but also being aware of sensitive issues and personal preferences—requiring

excellent social skills. This is a very important task, as a Russian developer (living in Germany at that time for a year already) explained. As an example, he reported about an incident where the Russian team had set up a homepage in a rush, following a request of a potential Russian customer, without consulting the German team. The new homepage included, besides internal technical information (for example IP addresses of internal services like the CVS), copyrighted images "borrowed" from the Internet. The German team demanded the immediate removal of the homepage, but the Russian team seemed to lack any understanding of the legal problems their action could entail. They reacted asking: "Why are you starting a war on this?" According to the Russian developer living in Germany, this had to do with the prevailing "culture of blame" in Company A (as labeled by our informant), which affected the interpretations of the event in a negative way—and he had many difficulties in moderating the conflicts arising from this incident.

In Company B, the Romanian manager played a key role in running the company; her seven 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 had permanently an open channel with the other manager and with the Romanian developers. During the interview, the Irish manager spoke about how collaboration with Romania would have probably 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 had given him confidence in her skills and commitment, and consequently in the people she recruited in Romania.

## 6. Discussion

In the following, we will examine the role played by culture in the knowledge management practices described in the previous section. For each, we will examine the impact of culture as reflected in practices, artifacts and routines. As mentioned previously, culture is an elusive concept that can only be studied by referring to its manifestations.

Regarding the first category of practices, **Status Meetings and Maintaining Awareness**, our observations reveal how onshore organizational culture can be extended to the offshore location. In Company A, both sites have their own weekly status meetings, but only the Tomsk site has to write minutes in English and send them to Bonn, the team in Bonn not having to reciprocate. In Company B, the joint practice of the two managers of having status meetings every morning is complemented by the maintenance of open Skype channels throughout the work day between managers, managers and developers, developers and customers.

Regarding knowledge exchange, the primary role of meetings and supporting artifacts (meeting minutes) is maintaining awareness. However, these meetings are also an opportunity for people to get to know each other as they are used as props for requests and communication between the teams. Thus, they also allow differences to become apparent.

**Collaborative Use of Shared Artifacts and Repositories.** The role of artifacts and repositories is to support collaboration. But team members from onshore and offshore might have conflicting understandings of the role of specific artifacts, the importance to keep them up-to-date, who should contribute, who benefits. The daily exchanges of technical knowledge between onshore and offshore observed were more or less unstructured, highly situated and

bound to emerging work trajectories, for example when unexpected problems occurred, or if changes in one part of the code base affected other modules. The onshore and offshore teams used shared repositories and communication channels, but as Krishna et al. showed [26], this cannot guarantee the success of collaboration in any way.

Regarding specification documents, the practices are quite different in the two companies, reflecting differences in the organizational cultures regarding customer relationships management. While in Company A the relationships with the customers are handled exclusively by the German employees, in Company B this responsibility is delegated to the Romanian developers themselves. One interesting detail about this is that the communication between the German and Russian site happens in English—none of the teams actually using English for intra-team communication, while in the case of Company B, the usage of English is prevailing, developers having to work directly with their international customers. As far as Company A is concerned, the practice has clear consequences for the teams, who have to deal with translations and/or communicate in a foreign language in large parts of their daily work. This also had impact on the organizational culture, concerning the routines of handling specification documents and the mediation of the contact between the Russian developers and the German customers.

Although the case of the Russian team working on obsolete features is an extreme example, it illustrates how particular organizational practices look from each side's perspective. When problems occur, the differences tend to be attributed to the national culture ("Germans are like this, Romanians are like that!"), although many might be due to the specific division of tasks (e.g. business analysts vs. developers). As the Russian developers of Company A had no direct contact to the customer, they needed clear and detailed instructions. Their German counterparts, on the other hand, worked under totally different circumstances, and needed to keep a close connection with the customer. Hence, they preferred to work in an agile way, with requirements being subject to ongoing negotiation and change. Existing studies have showed that most developers prefer to write code and not documentation [27,28], as was also confirmed by the project manager of Company A. In a similar fashion, the tendency toward innovation and adding state-of-the-art details, as opposed to working on the features stipulated in their contract is the object of another well documented dispute between developers and project managers, irrespective of nationality [24].

Referring to the practice of **Spending Time at the Other Site**, in Company A, where the visits of Russian developers in Bonn are relatively long term, the purpose is clearly defined: enculturation, knowledge exchange, transfer of particular technical skills to German team members. In Company B, the visits are relatively short, and are meant to give employees the opportunity to see the environment and the people they are collaborating with, get a direct experience of what is like to work from there. On the other side, these visits are also useful for perceiving and understanding the differences between sites. The existence of social ties has been shown to improve knowledge transfer and communication in general [29].

The practices of creating and maintaining social ties were closely related to visits to 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 attributed by our interviewees to cultural differences.

In connection with the practice we named **Mediating between People and Cultures**, we found that social connections and human 'bridges' between the teams played a very important role for knowledge exchange (our findings being pretty similar with those of Milewski et al.

[30], Cataldo et al. [31], Marczak et al. [12], Krishna et al. [26]). It became apparent that companies in both our cases rely heavily on some key people (called “liaisons”, “bridgeheads”, “straddlers”, “cultural mediators” by different sources), who act naturally as information brokers and conflict mediators. The “bridging” roles were mostly informal and taken on voluntarily, these people understanding the different perspectives of both teams and doing their best to fill the gaps. Hence, they contributed to the creation of shared understandings. These ‘bridges’ were also very important for the continuous reorganization of the shared practices, and they contributed to the mediation of “team knowledge”, found to be very important for efficient teamwork in distributed software development by other studies [32]. More specifically, the knowledge exchange between teams was heavily reliant on relationships between particular developers who possessed excellent social networks in both teams. In order to develop software in distributed teams, companies need to establish shared practices and ways of dealing with the everyday’s work or what Krishna et al. [26] name a “compromise work culture”, that is permanently re-negotiated and adapted.

The practices we described were based mostly on direct or mediated communication between people, and less on tools, repositories and artifacts. Instead of pursuing standardized and formal knowledge management approaches, knowledge exchange amongst practitioners was informally performed, based on triggers found in weekly meeting minutes or development databases—which also served as props for initiating context-related coffee/lunch discussions. This also applied to the knowledge exchange between the teams, where our field studies showed an almost complete absence of traditional knowledge management strategies, and a high reliance on informal communication via Instant Messenger or face-to-face meetings.

In this regard, it was interesting to see that the practitioners in our study mostly referred to national differences in breakdown situations, and that the use of cultural interpretations was often connected to negative interpretations and mutual criticism. Our observations coincide with the findings of Damian and Zowghi [33], who found that national culture was sometimes used as a “scapegoat” to explain ineffective management practices at various levels. More specifically, in our case studies, national interpretations were often used in terms of (over-simplified) stereotypes from an external point-of-view, when the behavior of the other team was considered to be inappropriate or unprofessional. The actors used stereotypical references to national culture in situations where professional or organizational explanations did not seem to work anymore, i.e. in conflict situations, emphasizing the “us vs. them” delineation between teams. In this regard, cultural interpretations were used to label the others when the collaborating teams lacked shared understandings, generating affirmations such as: “This is how the Russians (or Germans) are like!”

## **7. Conclusion**

Our stated aim for this study was to examine what we can learn from these two cases, considered by our informants as successful and long lasting collaborations, despite the occasional misunderstandings and frustrations (see figure 2 for an overview).

From what we have seen in our two cases, the organizational culture is permanently re-negotiated and adjusted to fit the distributed collaboration; the influences of national culture and site-specific organizational culture are subtle and not easy to separate from other factors. Socialization and travel between sites, although they come at a cost, have lead to good results and the two companies in our study believed they are important elements of their collaboration.

One important conclusion is that like any other collaboration, distributed collaboration is a continuous learning process: teams and people learn, becoming aware of the cultural differences, and adapt.

The cases presented in this paper cannot be, by any means, considered representative, and our aim was to rather to illustrate the intricacies of knowledge work in a distributed software development setting than to attempt any large-scale generalizations. However, we do wish to point out how a recourse to explanations based on national cultural differences are often, on closer examination, shown to have other rationales which do not use such a simple conceptual framework. We have shown how incidents usually attributed to national culture are mostly due to the incapacity to explain the other party's behavior within one's own reasoning frameworks (Nicholson and Sahay [4] have reported similar findings with regard to knowledge embeddedness).

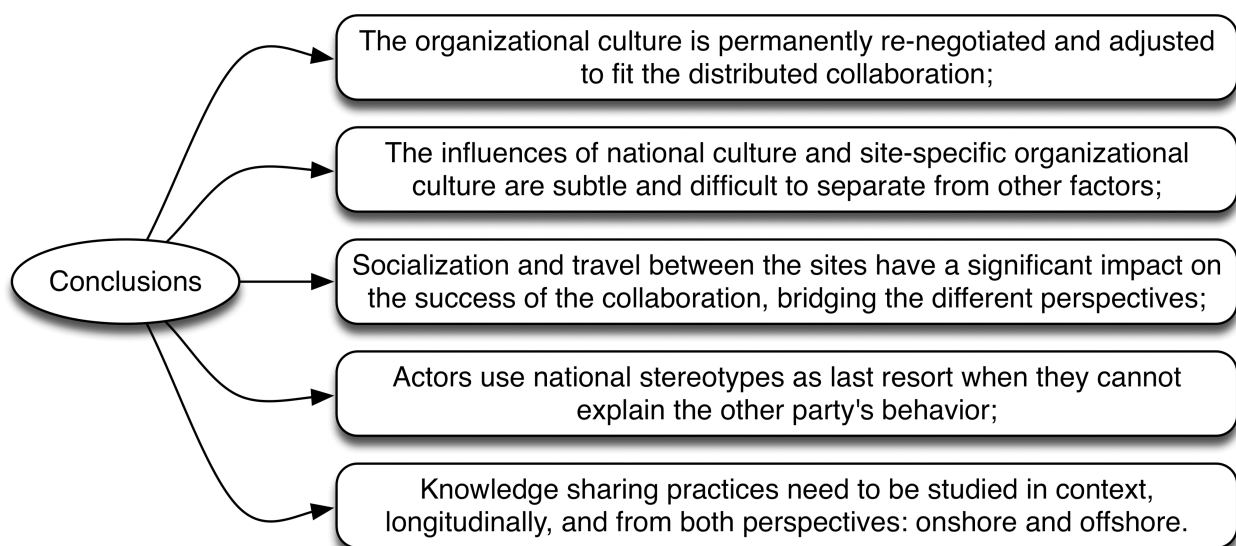


Figure 2. Lessons learned.

From a methodological point of view, we insist on the importance of looking at practices in context, paying attention to the correlated and embedded motivations, maintaining a dynamic perspective. Taking into account both the onshore and offshore perspective is vital for gaining a real understanding of complex situations (as we have seen that the perspectives can significantly differ between the cooperating teams). Our argument here can be stated as follows. In studying distributed software development practices, the concept of culture needs to be addressed, not simply in terms of national stereotypes, but through examining aspects of professional and organizational as well as local cultures. Investigating such issues is a difficult process, and requires an appreciation of the strengths and weaknesses of different methodological approaches. What we have attempted to illustrate here is one way of getting into the field to study such questions. No one approach will provide all the answers. However, we believe that the predominance of studies in GSE that apply survey-type instruments utilizing forms of Hofstede's dimensions of cultural variation is problematic for the field, as it tends to mask the value of other, more interpretive approaches to understanding the myriad ways in which communication

and coordination across globally-distributed software teams is affected by local, organizational and professional “cultures”. It is also worth emphasizing that the application of specific research methods and techniques is not separable from an understanding of the conceptual frameworks within which such instruments have been constructed. One cannot simply take instruments developed within a particular framework and apply them without a solid grounding in that particular approach. There is a danger within software engineering of researchers applying various forms of social science methods and analyzing the “results” without such a grounding. (This holds true across-the-board, and not only to studies using Hofstede’s frame). This issue requires more extensive deliberation and discussion within the SE research community itself, but it obviously also affects SE practice, in that the interpretation of many research studies in this area for practice need to be treated with a certain amount of caution, in terms of the generalizability of the findings, and their applicability to particular settings.

## 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) under PI grant 03/IN3/1408C .

## 9. References

- [1] F.O. Bjørnson und T. Dingsøyr, “Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used,” *Information and Software Technology*, vol. 50, Okt. 2008, S. 1055-1068.
- [2] H. Timonen und M. Jalonen, “A Critical Review of Knowledge Management Literature: Introducing a Practice-based Approach on Knowledge Sharing,” *9th Europaeen Conference on Knowledge Management*, Southampton, UK: 2008.
- [3] M.S. Ackerman, V. Pipek, und V. Wulf, *Sharing expertise: beyond knowledge management*, Cambridge, Mass.: MIT Press, 2003.
- [4] B. Nicholson und S. Sahay, “Embedded knowledge and offshore software development,” *Information and Organization*, vol. 14/4, 2004, S. 329-365.
- [5] Y. Hsieh, “Culture and Shared Understanding in Distributed Requirements Engineering,” *Proceedings of the IEEE international conference on Global Software Engineering*, IEEE Computer Society, 2006, S. 101-108.
- [6] G.H. Hofstede, *Culture's consequences: comparing values, behaviors, institutions, and organizations across nations*, Thousand Oaks, Calif.: Sage Publications, 2001.
- [7] L.A. Suchman, *Plans and situated actions: the problem of human-machine communication*, Cambridge, New York, Port Chester, Melbourne, Sidney: Cambridge University Press, 1987.
- [8] M. Huysman und D. de Wit, “Practise of Managing Knowledge Sharing: Towards a Second Wave of Knowledge Management,” *Knowledge and Process Management*, vol. 11/2, 2004, S. 81-92.
- [9] M. Huysman und V. Wulf, *Social capital and information technology*, Cambridge, Mass.: MIT Press, 2004.

- [10] W.J. Orlikowski, "Knowing in Practice: Enacting a Collective Capability in Distributed Organizing," *ORGANIZATION SCIENCE*, vol. 13, Mai. 2002, S. 249-273.
- [11] M. Granovetter, "Economic Action and Social Structure: The Problem of Embeddedness," *American Journal of Sociology*, vol. 91, 1985, S. 481-510.
- [12] S. Marczak, D. Damian, U. Stege, und A. Schröter, "Information Brokers in Requirement-Dependency Social Networks," *Proceedings of the 2008 16th IEEE International Requirements Engineering Conference*, IEEE Computer Society, 2008, S. 53-62.
- [13] V. Casey, "Leveraging or Exploiting Cultural Difference?," *Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on*, 2009, S. 8-17.
- [14] M. Gallivan und M. Srite, "Information technology and culture: Identifying fragmentary and holistic perspectives of culture," *Information and Organization*, vol. 15, 2005, S. 295-338.
- [15] O. Gotel, V. Kulkarni, M. Say, C. Scharff, und T. Sunetnanta, "Quality Indicators on Global Software Development Projects: Does "Getting to Know You" Really Matter?," *Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on*, 2009, S. 3-7.
- [16] K.C. Desouza, Y. Awazu, und P. Baloh, "Managing Knowledge in Global Software Development Efforts: Issues and Practices," *IEEE Softw.*, vol. 23, 2006, S. 30-37.
- [17] 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, S. 1-5.
- [18] 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, S. 243-258.
- [19] 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, S. 89-118.
- [20] C. Geertz, "Thick Description: Towards an Interpretive Theory of Culture," *The Interpretation of Cultures: Selected Essays*, C. Geertz, hrsg., New York: Basic Books, 1973, S. 3-30.
- [21] A. Boden, B. Nett, und V. Wulf, "Coordination Practices in Distributed Software Development of Small Enterprises," *International Conference on Global Software Engineering*, 2007, S. 235-246.
- [22] A. Boden, B. Nett, und V. Wulf, "Operational and Strategic Learning in Global Software Development - Implications from two Offshoring Case Studies in Small Enterprises," *Software, IEEE*, vol. PP, 2009, S. 1.
- [23] G. Avram, "Developing Outsourcing Relationships: A Romanian Service Provider Perspective," *The First Information Systems Workshop on Global Sourcing - Services, Knowledge and Innovation*, Val d'Iserre, France: 2007.
- [24] I. Richardson, G. Avram, S. Deshpande, und V. Casey, "Having a Foot on Each Shore - Bridging Global Software Development in the Case of SMEs," *Proceedings of the 2008 IEEE International Conference on Global Software Engineering*, IEEE Computer Society, 2008, S. 13-22.
- [25] G.M. Olson und J.S. Olson, "Distance Matters," *Human-Computer Interaction*, vol. 15, 2000, S. 139-178.



- [26] S. Krishna, S. Sahay, and G. Walsham, "Managing Cross-Cultural Issues in Global Software Outsourcing," *Communications of the ACM*, vol. 47, 2004, S. 62-66.
- [27] B. Nicholson und S. Sahay, "Some political and cultural issues in the globalization of software development: case experience from Britain and India," *Information and Organization*, vol. 11, 2001, S. 25-43.
- [28] T. Lethbridge und J. Singer, "How Software Engineers Use Documentation: The State of the Practice," *IEEE Software*, vol. 20/6, 2003.
- [29] J. Kotlarsky und I. Oshri, "Social Ties, Knowledge Sharing and Successful Collaboration in Globally Distributed System Development Projects," *Europaeen Journal of Information Systems*, vol. 14/1, 2005, S. 37-48.
- [30] 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, S. 23-32.
- [31] M. Cataldo und J.D. Herbsleb, "Communication networks in geographically distributed software development," *Proceedings of the ACM 2008 conference on Computer supported cooperative work*, San Diego, CA, USA: ACM, 2008, S. 579-588.
- [32] J. Espinosa, S. Slaughter, R. Kraut, und J. Herbsleb, "Team Knowledge and Coordination in Geographically Distributed Software Development," *J. Manage. Inf. Syst.*, vol. 24, 2007, S. 135-169.
- [33] D. Damian und D. Zowghi, "An Insight into the Interplay between Culture, Conflict and Distance in Globally Distributed Requirements Negotiations," *Hawaii International Conference on System Sciences*, Hawaii: 2003, S. 19-29.