Of Deadlocks and Peopleware - Collaborative Work Practices in Global Software Development

Gabriela Avram
Interaction Design Center, University of Limerick and
Lero - the Irish Software Engineering Research Center
gabriela.avram@ul.ie

Abstract

As part of a research project dedicated to the Social Organizational and Cultural Aspects of Global Software Development, the author has chosen to focus on collaborative work practices and knowledge management aspects of collaborative work. More precisely, the focus is on how the global distribution of software development affects collaborative work.

The current paper is a first attempt to unveil, through a concrete situation observed in a distributed software development environment, the complex ways in which people use technology to establish collaborative work practices.

By using ethnographically-informed methods, the author presents a bottom-up study of actual work practices, meant to contribute to a better understanding of collaborative work and knowledge management processes in distributed software development.

1. Introduction

Global Software Development is part of the globalization phenomenon, made possible by information and communication technologies. The accounts about cheaper work and "follow-the-sun" approaches are fading, while factors like proximity to the markets, access to specific expertise, productive friction and innovation capability tend to take the lead in driving the trend toward Global Software Development. While developments like Service Oriented Architecture and Web 2.0 seem to favor future GSD initiatives, the increased complexity of work generates difficult coordination, communication and collaboration problems.

Global distribution is adding new types of challenges for software development: strategic, cultural, and technical issues; knowledge, project, and process management; and inadequate communication[1].

From our perspective, software development must be seen as a human, social and organizational activity as well as a technical activity. Software development is also a knowledge intensive activity [2]. There seems to be an increasing awareness of the importance of human factors, especially in the Knowledge Management literature[3] and a shift toward an understanding of work practices [4].

People are probably "the" most important factor when trying to improve activity in this domain, and this is what Tom DeMarco and Timothy Lister tried to emphasize in naming their book dedicated to project management in software development: "Peopleware" [5]. Even if at the time, the distributed aspect wasn't salient, DeMarco and Lister claimed that the major problems were "not so much *technological* as *sociological* in nature." While software and hardware are made of replaceable modular components, software development teams are not.

This need to address organizational and social issues becomes even more critical in Global Software Development [6], where communication, coordination and control processes are also affected by geographical, temporal and cultural distance.

Our work is an attempt to complement top-down prescriptive approaches with a bottom-up study of actual work practices, in order to achieve a better understanding of collaborative work and knowledge management processes in a distributed software development environment.

In this paper, we focus on collaborative work practices in a specific global software development setting. More precisely, we are considering the following research question: How does the global distribution of software development affect collaborative work practices in a particular work setting? What can we learn from the particular

International Conference on Global Software Engineering(ICGSE 2007)
0-7695-2920-8/07 $25.00 © 2007 IEEE

challenges and the ways people use for coping with them in this specific situation?

The next section includes a description of the wider context of our work. Section 3 describes the case of a software development team (CROWOLF) from the Irish site of a multinational company. Section 4 is focusing on a concrete situation from the system testing phase; a teleconference with participants from two sites situated in Ireland and in the US. Section 5 discusses the collaborative work practices observed in this specific setting, with an emphasis on the impact of distribution, while the final section presents the conclusions.

2. The context of our research

2.1 The socGSD project

At the University of Limerick, Ireland, a group at the Interaction Design Centre has received national funding as part of a software engineering research consortium to study the social, organisational, and cultural aspects of global software development (the short name of the project is socGSD). This project aims to explore through case studies, how organizations attempt to manage the coordination of engineering work via a variety of mechanisms, from the formation of closely-knit, though distributed, teams through to the use of outsourcing, which still requires project management skills in order to ensure the quality of the output, and the integration of the output into the overall product. In our approach, we build on earlier Computer Supported Cooperative Work (CSCW) studies dedicated to issues such as articulation and coordination work, information sharing and knowledge management practices in distributed work, and the role of organizational memory support tools. The socGSD project is investigating the nature of these issues through a variety of analytical and empirical methods highlighting both theory and practice in this domain.

Over the past year, the socGSD team has been engaged in field work in several sites in Ireland where software development is being conducted involving geographically - distributed teams. Our research methods mainly rely on an interpretive, naturalistic approach to data collection and analysis. This means that we study the phenomenon in the actual settings where the work activity takes place, attempting to make sense of the work through the eyes of those actually doing it. There are a variety of methods which are being employed in this kind of qualitative research – interviews, both individual and group, introspective reports and diary and story analysis, observational studies, including shadowing of people during the course of their activities, analysis of documents, and workshops.

We attempt to bring into light the diversity of ways in which distributed teams shape their work practices, and achieve a shared understanding of their objectives. Another topic of interest is constituted by the ways in which people involved in software development manage to cope with the organizational rules and formalizations they are required to use through their work practices, even when these, at times become real barriers in accomplishing the work.

2.2 Research approach

What differentiates our approach from other GSD studies looking at distributed work arrangements are the extended field studies of workplace activity we are undertaking. Our work is informed by a long series of workplace studies presented in the CSCW literature.

We believe our approach complements studies undertaken from a macro-economic or strategic perspective or focusing exclusively on process improvement, by looking at participants in real workplaces in various GSD settings. These participants are studied in real work circumstances (as opposed to experiments), and are engaged in continually evolving working arrangements.

Our data collection and analysis methods are mainly informed by ethnography and include observation, document analysis, in-context interviews, audio recording, focus groups and workshops.

In the current paper, the author illuminates a number of work practices that were established in time by the members of two teams located respectively in Ireland and the US, using narrative as vehicle for research. The practices under discussion were not prescribed by the organisation (even if the existing organisational culture and available tools/infrastructure have definitely influenced them!), but devised by the collaborating teams to fit their current situation and needs.

2.3. The fieldwork at the described site

One of our field sites was the Irish subsidiary of a multinational company involved in software development. Following to a number of preliminary meetings with the company's management, where we explained our approach and negotiated access to the site, we were granted the opportunity to observe the activity of a software development team in January 2006. The team was consulted and agreed to participate

in our study. In March – May 2006 we participated in a number of team meetings, studied the documents in the project repository and interviewed the team manager and a few developers, trying to familiarize ourselves with the context and the work being done.

From June 2006 until January 2007, we have spent more than 40 days in the field, observing the activity of the team in its own work environment, participating in meetings and group activities and occupying a desk in the open plan next to the team's area. The periods of time varied between 6 consecutive days when the team was approaching a milestone or release, to 1 day weekly in order to maintain contact and awareness.

The research team was granted access to the company intranet, to the project document repository and its members were added to the team's mailing list. The author was also able to install and use the instant messaging system used by the developers on the team and to add them (and some of their contacts) on her contact list. This allowed the researcher to reach a better understanding of the ongoing activities and to continue the observation even when not present on site.

During our periods of observation, we developed a good relationship with all the team members and were given the chance to conduct both formal interviews and informal discussions on various topics.

One of the most important opportunities to observe directly the interactions with people in various other locations (US, Germany, India) was the author's silent participation in teleconferences. The team manager, who granted us this opportunity, was extremely supportive by not only allowing us to watch what was displaying on his screen, but also by answering to our questions at the end or commenting for us whenever possible. The remote participants were made aware of our presence in the room on these occasions.

The author kept a diary and took detailed notes on every day spent in the field.

In some of the group meetings, audio-recordings were allowed; on most occasions, these recordings were later put at the disposal of the development team for documentation purposes. In addition, some of the interviews were also audio-recorded. On a few occasions, permission was granted to take pictures of work spaces, meeting rooms, and teleconferencing equipment.

The author also traveled to the company's site in Germany and interviewed five people with different roles in the collaboration between the two sites (managers, architects, technical planners).

In August 2006, the research team organized two workshops (one with the development team and the quality engineers, and the other with managers and some of the remote collaborators of the team). The purpose of these meetings was to discuss our research-in-progress and illuminate some of the topics and situations we found of interest.

The data collected from the field was periodically discussed and analyzed by the extended research team, in order to identify topics, trends and problems and to compare the findings to those from other similar sites where fellow research team members were observing similar processes and activities.

3. Our case: CROWOLF

The CROWOLF project started in January 2005 with 15 developers, two software architects (one on site and the other one working from his home in Germany). Five of these developers had worked together before (lead by the same team leader) on a specific component of a similar product, bringing a valuable expertise to the team. A team of four quality engineers started its work on the product and collaboration with the developers later that year.

The team leader was subordinated to both a local manager and to a second one, located on the East Coast of the US. The team was also allocated a user interface designer and a technical writer, both located in US.

Several changes occurred over time. At the moment we are describing in this paper, the core development team (collocated in Ireland) included 11 developers, the software architect in Germany, a new UI designer and the technical writer located on the Eastern Coast of US.

A postmortem analysis was organized in August 2006 with the purpose of listing what the developers found to be the challenges and the achievements of the previous release and to learn from these. The author assisted in the preparation of this analysis and was invited to observe the event.

4. The deadlock

In this section, we focus on a particular phase in the lifecycle of CROWOLF: system testing. After presenting the broader organizational context and the context of the project, we use a narrative to present a specific snapshot from the activity of the team.

Narrative provides a way of organizing the complex forms of experience in ways which can be told and recounted. In the study of collaborative work practices in distributed settings, we need to pay attention to cultural and social processes and to contextual understandings. Narratives can provide useful insights by allowing us to build a rich picture and reflect the complex ways people chose for interacting through and with technology. The situation presented here is a common one: similar challenges

occur almost every day in this environment. What we aim to illustrate here are the complex and various causes that influence collaborative work practices in this distributed environment and the decisive role of people in this process.

4.1. The system testing phase

The situation described in the next section was not the first one when the CROWOLF team was under time pressure. In January 06, the company took the decision that the product will have to ship in July. With all the efforts of the team, their product was released, but only as a preview. A preview release is meant to demonstrate work-in-progress to the customers; customers can choose to try it at their own risk. Feedback from customers is highly encouraged, but the product is not supported. The good part is that the application gets some exposure and feedback, the bad part is that it doesn't produce any income for the company.

Because the software had only undergone functional testing on a Windows platform at the time, it was only released for Windows. After going through a successful cycle of long run system testing, the product was to receive full support and to be actually offered to the customers.

The development team in Ireland, the collocated functional testing team, and the US system testing team had a good working relationship; they were meeting regularly in conference calls, and they were using frequent email exchanges to update each other and to maintain group awareness.

The Irish team was known across the organization for taking initiatives like setting up their own server or setting up a similar computer cluster – as a way to speed up the debugging efforts. This gained them the admiration of other teams they worked with (as revealed in an interview with a German counterpart).

After going through extensive functional testing in the last 9 months, CROWOLF entered the system testing phase. While functional testing looks at how well the system executes the functions it is supposed to execute—including user commands, data manipulation, user interface, and components integration, the system testing phase is supposed to cover several platforms, running different operating and database management systems, and aiming at performance issues as well – load, volume, stress.

The functional testing phase had now ended. For this phase, the CROWOLF team had its own quality engineers at the Irish site. The system testing task was assigned to a US team, who had the required expertise and resources (involving several computer clusters running on different platforms).

Let us now introduce the people involved in the scene we intend to present.

On the Irish side, Pat is the development team manager for CROWOLF and Claude is the lead developer for the release undergoing system testing. Sean is the Quality Assurance lead for the functional testing and Ian is one of the three Quality Engineers(QE) in his team. Their main contact on the American site is Matt, the leader of the team of QEs involved in the system testing phase of CROWOLF.

Pat - the development team manager, and Sean – the QE lead, have been part of the organization for 10 years or more. Pat had run several projects before this one and gathered a lot of organizational wisdom. Even if he's not involved directly in coding, he has a thorough understanding of the technologies deployed. According to our observations, he is very good at planning and multitasking in a rapidly changing and extremely demanding environment; he also has good facilitation skills; documenting facts and decisions as he goes along has become a sort of second nature for him. He prefers to motivate and support his people rather than to use his authority, coming across like a dedicated and considerate person.

Claude is one of the most experienced developers on the team. In the earlier stages of the project, he volunteered to take up the responsibility for integrating the different components. His experience in this position equipped him with a thorough understanding of the product and of its operation. All his colleagues see him as an excellent team player, always friendly, always helpful; with all the stress brought in by pending deadlines, he was the one to point out that he was becoming a bottleneck and initiated knowledge transfer sessions for his team mates.

The Irish team has generally a very good impression of Matt – the QE lead for system testing- and his people; Matt comes across as a very competent guy, hard working, showing a lot of dedication and sacrificing his personal time for solving the testing problems. Matt usually starts at 6 am EST to increase the overlap time with the Irish team, but he also works on week-ends and nights from home. He seems passionate about what he's doing, but on the Irish side there are concerns that sometimes exhaustion might make him more prone to error.

In July '06, Claude(as lead developer) and Ian(one of the four QEs of the Irish team) went over to the US site to support Matt and his team to set up the computer clusters needed for this testing. It was emphasized on several occasions that a lot of persuasion was needed to get this trip approved, but both parties had considered it necessary.

The visit was prepared during a first online get-together of the developers and QEs involved in functional testing from the Irish site and US-located QEs in June '06. A previous testing plan drafted in February '06 was brought up-to-date, and the functional testing experience to date considered a valuable input for the next phase (Fig.1 presents a timeline of the collaboration between the two sites).

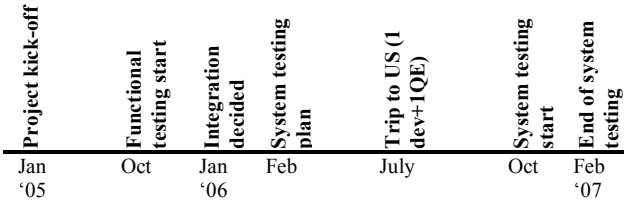

Fig.1 The CROWOLF project timeline

During their 2 weeks in the US, Claude and Ian helped set up a cluster of computers documenting each and every step as they went on. This enabled them to and their US counterparts to develop a fair understanding of each other's competencies and working style. Good personal relationships were established, and they also had a chance to compare their own work environment with the US site(as revealed in an interview with Claude after his return).

One of the US team's tasks was to write the scripts to automate the testing process. This (as emphasized by the second line manager in a conference call) couldn't have been done without the deeper knowledge on CROWOLF's intended use and functioning possessed by their colleagues from Ireland. Back in Ireland, the two considered the trip as a success, but warned their colleagues during the next weekly meeting that "the difficult part" of the testing process was "yet to come".

Most of the daily communication takes place through e-mail, instant messaging, phone and conference calls. Transactive memory systems [7] are available in different shapes (expertise browser, white pages, social networking applications), but people (and groups) tend to rely mostly on their personal contacts when it comes to find or recommend an expert in a specific matter. The use of instant messaging gives them then to opportunity to get in touch with any colleague around the world, in the shortest time possible.

According to our observations, there are frequent situations when people situated in remote locations have to synchronize their work (waiting for a bug fix before the test procedure can continue, having a phone conversation, sharing screens or looking at the same document or code component in a repository in order to discuss a problem). A scheduled update of the computer cluster in the US required that the people accessing them remotely from Ireland or Germany 'take their hands off' them. Remote tests ran by team members in Ireland were normally allowed, but they were subjected to follow a specified coordination procedure. A hard disk crash, a power shortage or a change in the office configuration that involved moving computers around affected people working in remote locations as much as the local team.

Despite the emergency situation described below (that became, in a way, permanent!), people in all sites had to go on with their lives. There were weekends and there were holidays, family obligations and family trips. The common practice was to give the others notice about time intervals or dates when they were planning to take time off and to indicate contact details for emergencies (mobile phone numbers, e-mail). This kind of coordination proved effective, the team in the US making the effort to work on week-ends whenever needed to get a new round of tests started and gain time. Checking the status of the testing from home during week-ends was another common practice. In one extreme case, a member of the Irish team had to join a meeting for consultation while abroad on vacation.

In the following vignette, we present a snapshot of one particular instance meant to illustrate collaborative work practice in this particular setting.

4.2. A snapshot: the status meeting

It is Tuesday morning, 8am EST, 1pm GMT (October 2006) - early in the morning in the US and lunch time in Ireland.

On the Irish side, there are three participants in the meeting room: Pat, Claude and Sean. A fourth participant, Ian, is working from home today, so he is taking the call from there. Sean initiates the meeting by dialing into the conference system.

The meeting was scheduled in advance by Pat, who also sent the invitations and set up a conference call. The conference system informs them there are three parties online, including them. The participants greet each other and introduce themselves.

Figure 2 pictures the setting. There are three participants from the US site in the meeting, but Matt was the only one active most of the time. On the Irish site, there are three participants in a meeting room (joined by the researcher observing the meeting), and one participating from his home. Pat and Claude are the ones who are leading the discussion. There are no participants from the German site in the meeting, however a possible involvement of Felix is repeatedly taken into consideration.

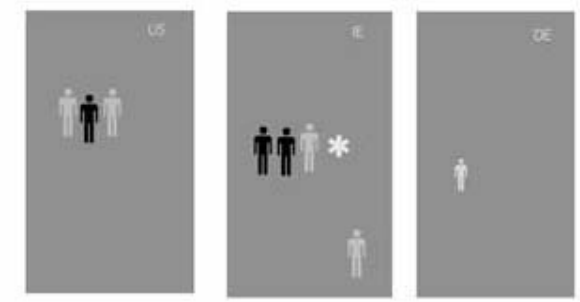

Fig. 2 Meeting participants

On the US side, the three QEs assigned for the system testing of CROWOLF are now all on line, probably from a meeting room resembling to the one here in Ireland. Matt leads the discussion. As the others had only minor interventions during this meeting, we will not name them here.

In the weekly team meeting one day before developers were warned that CROWOLF entered the system testing phase and the following three weeks will be critical. The team was asked to be ready to support the testing and bug fixing effort. Even if this phase only started a few days before, the team was already confronted with blockages.

The current meeting is about the deadlock blocking the testing efforts. They're stuck, and nobody seems to know what to do. They were already 9 days behind the schedule on Sunday, and no one can see any solution that could change the situation. This kind of situation is not uncommon in software development, where the high level of work complexity induces problems that are both difficult to foresee, and challenging to solve.

The computer cluster on which they are running CROWOLF under Windows is broken again, after an apparent success to unblock it on Saturday. Actually, during the weekly team meeting held the day before, three people had received congratulations for their contributions during the week-end (one in the office coordinating with two other working from home) that apparently solved the problem blocking testing.

It now looks like it was only an impression: *"We don't know what was wrong in the first place, how we fixed it, and now it came back!"(Claude)*

Three people are actually debating the problem: Pat, Claude and Matt. The rest of the participants are mostly listening in – they have to be aware of what is going on, in case they can be of help. They explore possible workarounds for a rapid unblock, but they all agree that the real problems have to be identified and solved, otherwise they will keep coming back.

Matt's tone lacks energy - he doesn't sound angry, but seems exhausted – he doesn't see any possible solution right now. "We're trying to do surgery here with a butter knife to get this working again!".

They review the three bugs they had identified, hoping for a new insight to show.

Possible courses of action are examined. Bringing in more people at this stage doesn't seem productive - Claude is already working on the issue full time, being unavailable for the rest of things going on – and there are plenty (two other releases are on the pipeline!). Matt, on the other side, has no other people available.

The testing seems stuck, and there's neither an obvious solution nor a time estimate for a possible unblock.

Pat: *Were we better a few weeks ago? The other cluster seemed to be working…*

Claude (obviously disappointed and trying to cope with a new perspective on the situation): *only seemed to be…*

The meeting goes on with the exploration of alternatives:

Matt: *maybe we should set it up again – the way we did when you guys were here!*

Claude uses this opportunity to sneak in a hint on his actual point of view: *the problem might be due to a configuration error!*

One of the QEs on the US side mentions he continues to document as he goes on with the setup and install, following the example of Claude and Ian during their visit in July.

Matt mentions a couple of Read.Me files circulated by Claude and recommends them to his colleagues.

As usual, the end of the meeting approaching, they try to devise some action items. A possible involvement of the developers at the German site, who developed both the enterprise portal and the runtime engine, comes into discussion.

Pat: *let's ask Felix what could be the cause for the enterprise portal behaving the way it does! Any hints? Is there anyone who tested this previously?*

Felix is one of their colleagues at the German site, part of the team who developed the enterprise portal and their preferred contact when it comes to portal set up and configuration problems.

Matt knows that the some of his colleagues in the US site were involved in the testing of the enterprise portal. Pat keeps on wondering if this is the first time when this problem occurs; he hopes that by contacting the German team, he could get a hint. Being the first in the organization who build a product dependant on the runtime engine (which, on its turn, is based on the portal technology), there might be a chance that the problem wasn't generated by the CROWOLF code itself, but raised by another malfunction in the runtime engine. But all these are just guesses!

Claude is nodding, showing his total approval for Pat's point.

Matt: *is there any other group using the runtime engine yet?!* No answer.

We're 10 days behind today; tomorrow we'll be 11 days behind and we're overlapping our end date; we're red, and we have no idea when we'll get green again! (Matt's tone reveals disappointment and frustration).

They seem to lack a solution, and lamentations do not help – so they decide to end the meeting at this point. Claude suggests having a 5 minute update daily, and everyone agrees.

The phone connection is terminated and Sean leaves the room, running to his next meeting or maybe trying to get some late lunch – but the discussion between Pat and Claude continues; there are things that were not spoken, and the two feel the need to share their impressions. They share a strong feeling that the problem might be due actually to a fault in setting up and configuring the computer cluster they're using for testing.

Claude: *We got it to work; we documented everything- there're lots of manual steps!*

And now it's broken, and we have to put the pieces back together! Maybe I'm getting paranoid here, but … We fix it, they break it! We already got it to work two times there, and two times here…

Their assumption is based on previous experience. They worry about the superficial understanding Matt has of the product. However, what they need now is a solution!

Pat: *would they allow us to help them deploy again? Organize a meeting - and ask Matt to share what he's doing?*

Claude: *You mean…us sitting on his shoulder and watching while he's doing it again? If you can get it…but I'm sure they won't like it!*

What they are thinking of is a fresh start, with Matt setting up and configuring the cluster again, while sharing the screen(s) with them and talking about what he is doing.

Claude expresses his frustration for spending the whole morning documenting one of the problems they encountered. They discuss assigning a second person to solve this incident – but the resources are terribly scarce at the moment, as the team is working on two other different releases and another milestone approaches. They go through the possible alternatives once more:

- applying a workaround - using the "portal knife"(a previously developed solution for temporary unblocking the cluster); it is only a temporary solution and it is lethal – no upgrade would be possible after that.
- a re-install they would be able to assist to ("sitting on Matt's shoulder" and monitoring his steps); it might solve the situation, but, as Claude mentioned, it could damage their relationship badly.
- getting support from the German team by getting Felix involved; Felix's team wrote the code causing the trouble when interacting with CROWOLF; they should know a workaround ("*they should have their own butter knife*".)
- flying someone over to the US location– either Claude or Ian; obviously, they will need approval for this and it might collide with their personal priorities at this moment.

Efforts are already ongoing at the Irish site to set up a local computer cluster there in order to try and reproduce the problem.

4.3. The solution and the next steps

Two days later, the cause was identified as a mundane omission of a link and the deadlock was solved. It wasn't actually Matt's failure in configuring the computer cluster, as Claude and Pat supposed. All the involved parties felt relieved. They were one step further!

The idea of setting up a parallel computer cluster to run their own tests resulted in more machines being made available for the Irish team.

When the long runs began to last more than 24h without getting blocked, the focus shifted toward performance improvement. Various statistics were circulated by Matt to both the Irish and to their German counterparts (who owned the underlying technology and discussed the progress of CROWOLF regularly). Developers on both the Irish and the German site were able to access the extensive logs of the long runs and look for patterns. Several solutions were found to improve both the performance of the runtime engine and that of CROWOLF.

In the following two months, the teams involved continued their efforts for getting the long run tests on CROWOLF done and declaring it ready for shipping. The allocated resources needed to be supplemented. Deadlock after deadlock, new bugs were identified and fixed until the code was stabilized. How one of the participants in this effort put it, they were on the "bleeding edge" of large-scale software development, where all the interdependencies show and constant cooperation between all the actors involved is irreplaceable.

Is this an illustration of a success, or of a failure situation? One member of the Irish team told us that "after working on so many projects, it can be difficult to see success sometimes". There are temporary

victories that need to be acknowledged, but there is no black and white distinction between success and failure when it comes to large software development projects.

5. Discussion

There are a few points regarding collaborative work practices we would like to discuss here. For each of them, we will look at how the distributed aspect of work influences it, and how people cope with it.

5.1. Knowledge management practices

5.1.1. "How to" – the situated learning aspect. Documenting is well-known as not being the favorite task of software developers [8]. In the scene presented, documenting "as you go" was mentioned as a way of capturing the details of the set up and configuration process and dealing with the complexity of the situation. During their stay at the American site, the two members of the Irish team who possessed the knowledge about CROWOLF made the effort to externalize this knowledge and turn it into a reproducible procedure. Collocation was considered absolutely necessary in this situation for knowledge externalization and transfer. On one hand, setting up the cluster was a trial-and-error process where direct interaction of the people who had the expertise with the computers was salient. On the other hand, the documentation they wrote only included the successful steps, but not the problems encountered. The best way to learn how to do something is by watching someone who can do it, and later on repeating the same steps [9]. This is exactly what one of the solutions envisaged: watching Matt reproducing the previously documented steps- aka "sitting on his shoulder". What Pat and Claude were worried about was not only that Matt might have missed or misunderstood one of the steps. They were mostly worried about not being able to pass all the required knowledge – actually the underlining "philosophy" of the product – to the American QE team. This was vital, because the set up and configuration procedure had to be properly automated and documented in order to be made available to the potential customers.

5.1.2. "Who knows what?" – the transactive memory aspect. Knowing who has a specific expertise and having access to that person is a vital aspect when it comes to complex situations like the one described above and time constraints. In the situation we described here, the CROWOLF team was building on top of two other technologies developed inside the company. It was logical to ask the developers involved in the two other projects if they ever encountered a similar problem. "Is there anyone who used this before?' and "Let's ask Felix" are two different illustrations of this aspect. In the first case, it was an attempt to tap into the transactive memory [7] of the QEs at the US site; Quality Engineers are, in a way, boundary spanners, because they have to deal with various systems designed by different teams distributed globally and acquire relevant information about who knows what. In the second case, they were certain that their German colleague could be able to help.

Looking for the right people across locations and across domains of expertise is mainly done through the personal contacts network in most of the situations we observed. Expertise browsers are useful once a person has been identified, but they are very seldom the starting point. Instant messaging -in the context of an organizational culture that encourages its use – provides people with a straightforward way to access the identified experts.

5.1.3. Awareness maintaining – the mutual knowledge aspect. In the case under scrutiny, people use a variety of practices for keeping the interested parties on the same page and for building a shared understanding of the issues. Regular meetings are one of the most frequently used. Because walking to each other's desk or meeting at the water cooler is excluded in distributed work settings, people tend to plan their interactions. Planning for a 5 minute daily meeting is actually creating the opportunity of having a brief update, where all the interested parties will be present and give or get the latest information on the critical situation. A synchronous interaction marked in people's calendars is untouchable ("the calendar is sacred!") and, according to our observations, is preferred to an email exchange when it comes to critical issues.

By circulating statistics and logs related to the progress of testing (or placing them in shared folders) and giving remote users access to the computer cluster running the application under scrutiny, the US team enabled the developers and architects to observe what was going on and develop their own analyses. This can happen in a collocated situation as well, but what was interesting in this case was that Matt had to ask his colleagues working remotely "to take their hands off" of his computer when he had to restart the computer cluster, or had to inform them about power shortages or malfunctions occurred.

The Read.Me files circulated by Claude and recommended by Matt to his colleagues constitute a good example for the way collaboration practices are actually established; documenting for maintaining awareness in a group has become a way to cope with the distribution of activities.

Maintaining awareness was a priority; it was essential in email exchanges to keep a wider circle of people aware of what's going on. In the case we described, selected people from the German team have been permanently kept in the loop with the progress on this particular testing process. This kind of peripheral participation maintains mutual knowledge [10] in a community of interest[11]. These people don't share a sole practice – there are developers, testers and managers at different levels involved, but they share a common goal. Instant messaging sessions between two individual are sometimes e-mail-ed to a whole group, because they contain relevant information.

Instant messaging also allows for social translucence [12]; people can get aware of each other's presence and take advantage of it when necessary. Members of the collocated team also happen to work from home, situations in which their interaction with the rest of the team is no different from the interaction with counterparts in the US and Germany – mediated by instant messaging, email, phone calls and remote access to project repositories. Our own observations in this direction were confirmed by several members of the team.

5.2. Adding more resources to the project

When it comes to additional physical resources, their location proved to be significantly important. The Irish team has applied for and received the approval to set up an own local computer cluster, in an attempt to mirror the problems encountered by the US QE team and solve them in the shortest time.

The situation is different in the case of people, where expertise is vital, location does not have the same importance. In the situation described, bringing more people in is one of the solutions taken into account. While asking support from people with expertise in the underlying technologies is favored, bringing another developer up-to-speed is not considered a worthwhile effort at this point. Documenting –not only vital for finding a solution, but also for sharing the context with other people and getting help – is, as Claude said, extremely time consuming and even frustrating.

5.3. The importance of social networks

In a distributed context, social relationships take a different dimension. In most cases, people do not get the chance to meet in person. The relationships are shaped by mediated interactions (participating in the same teleconferences, email exchanges, chats and phone calls), by peers' references and professional reputation.

Good communication is extremely important, as the situation described earlier illustrates. Simple features like good will and common sense prove to be vital for collaboration in an environment where everyone is under pressure and it is very easy to get caught in the blame game. A potential solution that could have caused a negative reaction in a counterpart who was so dedicated to the common goal was carefully considered and dismissed in the end.

Eventually, the deadlock was found to have a completely different reason. Initially estimated at few weeks, the testing efforts took several months and maintaining a good collaboration between the two sites proved paramount for the success of testing and bug fixing. People's interactions, their speed of reaction and capacity to self-organize and coordinate were vital in this dynamic context.

The social networks formed across locations and specialties proved to have an impact on how work is done (also confirmed by [13],[14]).

6. Conclusion

The purpose of our paper was to put into light a number of actual work practices through a particular case of collaborative work over distance.

We aimed at emphasizing the paramount role of human actors, their values and their social connections in getting work done, by illustrating the complex and various causes that influence collaborative work practices in this distributed environment. Some of the practices described here were developed by the collaborating teams as they went to fit their current situation and needs.

Our study is meant to contribute to a better understanding of collaborative work and knowledge management processes in distributed software development settings. This understanding constitutes a pre-requisite for better organization and a preliminary step in the design of the next generation of tools to support collaboration.

7. Acknowledgments

We wish to thank our informants in the observed company for allowing us to interview and live with them on site for many days, to all the members of the socGSD project in IDC.

This research was supported by Science Foundation Ireland under PI grant 03/IN3/1408C and by Lero – The Irish Software Engineering Research Center.

8. References

[1] J.D. Herbsleb, D. Moitra, "Global Software Development", *IEEE Software*, IEEE, March/April 2001, pp.16-20.

[2] I.Rus, M. Lindvall, "Knowledge Management in Software Engineering", *IEEE Software*, IEEE, May/June 2002, pp.26-38.

[3] Prusak, L., "Knowledge in Organizations (Resources for the Knowledge-Based Economy)", Butterworth-Heinemann, 1997

[4] Herbsleb J.D., Grinter R.E., Perry D.E., "The geography of coordination: dealing with distance in R&D work', Conference on Supporting Group Work, Proceedings of the international ACM SIGGROUP conference on Supporting group work, pp.306-315, ACM Press, 1999

[5] DeMarco, T, Lister, T., "Peopleware: productive projects and teams", Dorset House Publishing, New York, 1987

[6] F.Lanubile, D.Damian, H.L.Oppenheimer, "Global Software Development: Technical, Organizational and Social Challenges", *ACM SIGSOFT Software Engineering Notes,* Vol.28, No.6, Nov.2003

[7] Wegner, D. M. Transactive memory: A contemporary analysis of the group mind. In B. Mullen & G. R. Goethals (Eds.), *Theories of group behavior* (pp. 185-208). New York: Springer-Verlag,1986

[8] Lethbridge, T., Singer, J., Forward, A., "How Software Engineers Use Documentation: The State of the Practice." IEEE Software 20(6): 35-39 (2003)

[9]S.D.N. Cook and J.S. Brown, "Bridging Epistemologies: the Generative Dance between Organizational Knowledge and Organizational Knowing," in *Managing Knowledge*, edited by Stephen Little, Paul Quintas and Tim Ray, pp.68-101. Sage Publications, 2002.

[10] Cramton, C.D.," The Mutual Knowledge Problem and Its Consequences for Dispersed Collaboration", Organization Science, Vol. 12, No. 3, May-June 2001, pp. 346-371

[11] Wenger E., *Cultivating communities of practice: a guide to managing knowledge.* By Etienne Wenger, Richard McDermott, and William Snyder, Harvard Business School Press, 2002.

[12] Erickson, T. and Kellogg, W. A. "Knowledge Communities: Online Environments for Supporting Knowledge Management and its Social Context." Sharing Expertise: Beyond Knowledge Management (eds. Ackerman, Mark, Volkmar Pipek, and Volker Wulf). Cambridge, MA, MIT Press, 2003, pp. 299-326.

[13]Ehrlich, K. and Chang, K. "Leveraging expertise in global software teams: Going outside boundaries", Proceedings of the IEEE international conference on Global Software Engineering, ICGSE, 2006, pp.149-158

[14] Kotlarski, J. and Oshri, I., "Social ties, knowledge sharing and successful collaboration in globally distributed system development projects.", European Journal of Information Systems, Vol 14, 2005, pp 37-48.