



How a Geographically Distributed Software Team Managed to Negotiate Successfully using Chat Technology

Tenorio, Nelson; Bjorn, Pernille

Published in:
Revista Tecnologia e Sociedade

DOI:
[10.3895/rts.v15n37.8655](https://doi.org/10.3895/rts.v15n37.8655)

Publication date:
2019

Document version
Publisher's PDF, also known as Version of record

Document license:
[CC BY](https://creativecommons.org/licenses/by/4.0/)

Citation for published version (APA):
Tenorio, N., & Bjorn, P. (2019). How a Geographically Distributed Software Team Managed to Negotiate Successfully using Chat Technology. *Revista Tecnologia e Sociedade*, 15(37), 183-206.
<https://doi.org/10.3895/rts.v15n37.8655>

How a Geographically Distributed Software Team Managed to Negotiate Successfully using Chat Technology

ABSTRACT

Negotiation is best accomplished in collocated settings, and negotiation in geographically distributed settings is prone to failure with a risk of conflicts. Investigating distributed software development, we were surprised to discover that a software development team, located in different parts of Brazil, was able to negotiate successfully and reach an agreement to change from ticket-oriented processes towards release-oriented processes for bug fixing activities using only chat technology. In this paper, we explore how the chat technology allowed the distributed software team (including both vendor and client team members) to successfully negotiate and reach agreement about adopting and implementing a new collaborative workflow in the governmental IT-project. Our research method is based upon an ethnographically informed empirical study of the software development involved in a Brazilian software company. Thus, the data collected shows that the chat technology provided a platform for the team to engage informally in important discussions across locations. The chat technology allowed participants to navigate both within and across diverse subgroups (collocated client-developers; distributed client-developer, and distributed developers-developers), which supported successful subgroup dynamics avoiding the risk of conflicts emerging from faultlines.

KEYWORDS: Chat technology. Negotiation. Global software development. Distributed team. Double level language. Bug fixing. Faultlines. Subgroup dynamics. Software engineering.

Nelson Tenório
nelson.tenorio@unicesumar.edu.br
Instituto Cesumar de Ciência,
Tecnologia e Inovação (ICETI)
University of Copenhagen

Pernille Bjørn
pernille.bjorn@di.ku.dk
University of Copenhagen

INTRODUCTION

Software projects are often done in distributed settings, where clients and the software development team are geographically distributed. Despite the geographical distance participants often work in closely-coupled work arrangements (ESBENSEN; BJØRN, 2014; CRAMTON, 2001; JENSEN, 2014), structured by different types of agile methodologies (ESBENSEN; BJØRN, 2014; ŠMITE; MOE; ÅGERFALK, 2010). Such projects depend upon participants ability to navigate, coordinate, and communicate using diverse collaborative technologies (BJØRN et al. 2014; BJØRN; HERTZUM, 2006; BODEN et al., 2014; MARK et al., 2002) in which the majority of the interaction is accomplished, i.e., chat group, online forums, video conferences, document repositories, and emails (CHRISTENSEN; BJØRN, 2014; GUO et al., 2009; SEGENREICH, 2008; DABBISH et al., 2005; HERBSLEB et al., 2002). While the interaction in software projects is multiple and diverse, we, in this paper, are particularly interested in the negotiation activities within distributed software team.

Negotiation is a critical activity for software developers, where participants discuss and reach agreement about how and why certain details and structures are to be organized and implemented in certain ways and continue to be an activity throughout the whole project lifecycle (CHRISTENSEN; BJØRN, 2014). In geographically distributed settings, negotiations activities are facilitated and mediated by cooperative technologies (JOWETT, 2015; LI; ROSSON, 2014). However, technology-based negotiation activities have been identified as being prone to failure in geographically distributed settings. In this sense, researchers have pointed to working across time zones, culture, and professional language are some of the reasons for the challenges (MARK et al., 2002; OLSON; OLSON, 2000; VALLEY; MOAG; BAZERMAN, 1998). Given these insights from prior research, we were surprised to find that in our empirical case, where we studied a Brazilian software development team, that consisting of team members from both the vendor and the client managed to successfully negotiate. Moreover, that team implemented a new collaborative work structure using primarily text-based chat group technology.

Chat technology is of core interest to the Computer-supported Cooperative Work (CSCW) community, and the potential for using such technologies (e.g., Skype or Slack) in organizations of high complexity have been identified as an important research agenda (RIEMER; FRÖSSLER; KLEIN, 2007). Chat technology provides low-cost accessibility to team members across geography and time (MORAES; CABELLO, 2017; HSIUNG, 2000; ANDERSON; KANUKA, 1997). Moreover, chat technology can potentially facilitate closely-coupled interaction and communication within and across organizations (FAYARD; DESANCTIS, 2005; CLÉMENT; BAKER; MACINTYRE, 2003). By supporting 'lightweight' communication, chat technology provides alternative ways for participants to discover co-workers' availability, which potentially can trigger opportunistic communication supporting some degree of team context and facilitate cooperative inquiry to the entire team (HERBSLEB et al., 2002). Successful use of chat technology depends on participants abilities to establish and develop norms, context, common language, and problem definitions across all (MALHOTRA et al., 2001). However, negotiation activities –

especially cross-organizational negotiation, where financial and political considerations exist, make the opportunity to develop common language and shared context difficult to implement, and thus developing new technologies supporting negotiation across geography continues to be a challenge (BJØRN; HERTZUM, 2005; OLSON; OLSON, 2000). Therefore, this paper aims to explore how the chat technology allowed the distributed software team, considering both vendor and client team members, to successfully negotiate and reach agreement about adopting and implementing a new collaborative workflow in the governmental IT-project. In this sense, the research question we explore in this paper is: *How did the geographically distributed software development team successfully negotiate and establish a new workflow structure changing their work arrangement, using primarily group chat technology?* To answer that, we performed an ethnographically informed empirical study of the software development involved in a Brazilian software company, and we collected data observing chat groups which had fifty-five negotiation cases. Based upon our empirical findings, we find that the negotiation succeeds, not just because the team developed norms and common language, but because the chat group technology facilitated grounding activities (CLARK; BRENNAN, 1991) both within and across diverse sets of subgroups involved in the negotiation, namely client-developer at the same location; client-developer across location; developer-developer across location. When geographical distributed teams are composed of collocated subgroups, there is a tendency that such subgroups coalesce into smaller unites especially if the demographic attributes align with collocated subgroups, and such setups risk producing faultlines (CRAMTON; HINDS, 2004). We found that the software developers overcame the risk of faultlines in their negotiations, because of the affordances of chat technology allowed them to navigate across and within the diverse subgroups breaking down the barrier of demographic attributes and organizational belonging. The participants through cultural language exchange (ROBINSON, 1991) managed to create and navigate permanent records of decisions manifested through shared digital objects in the chat group technology. Thus, the chat group technology supporting synchronously interaction facilitating a dynamic negotiation context comprising of both informal and formal language exchange simultaneously.

The remainder of the paper is organized into six sections. Following by this introduction, we present the theoretical background of this study, then our research method, followed by the results of our analysis. Finally, we discuss our findings and provide our conclusion.

CHALLENGES FOR COOPERATIVE NEGOTIATION ACROSS GEOGRAPHICAL DISTANCE

Collaboration within geographically distributed teams is core concern for CSCW research, since its inception and there is a long cannon of research papers which have explored the challenge of distributed collaboration for the design of cooperative technologies in all kind of perspectives and in different domains (HINDS; RETELNY; CRAMTON, 2015; BODEN et al., 2014; OLSON; OLSON, 2000). One core domain for the research on distributed teams is software development (BJØRN et al., 2014), since geographically distributed software development has become the norm rather than the exception for how the work is organized when

we design IT systems (HERBSLEB, 2007). Core challenges for distributed software development have been identified as linking to temporal constraints (HERBSLEB; PAULISH; BASS, 2005), to coordination challenges (CHRISTENSEN; BJØRN, 2014), as well as to commitment and trust (SØDERBERG; KRISHNA; BJØRN, 2013). While the technological development has improved the conditions for distributed software development, one core challenge remains as to the challenge of creating common ground related to the project at hand as well as to how to collaborate (BJØRN et al., 2014).

Common ground is established through grounding in conversations, where participants provide evidence and references supporting their argumentation through aspects provided by the face-to-face shared context characterized by various aspects such as co-presence, visibility, audibility, and simultaneity (CLARK; BRENNAN, 1991). This means that whether it is possible to create common ground in distributed settings depends tremendously upon the affordances of the technology (e.g., chat, video conference, document repositories) supporting the interaction and the coordination of work (BJØRN; NGWENYAMA, 2009; HINDS; WEISBAND, 2003; ARMSTRONG; COLE, 2002; CRAMTON, 2001). Thus, to create common ground concerning the project and the process requires participants to have a fundamental basis, in this case, a shared context. That shared context can engage in the negotiations and discussions required to take important decisions facilitated by informal language constructs (ROBINSON; KOVALAINEN; AURAMÄKI, 2000). Finding ways to establish a shared context by which negotiation can take place supporting the distributed software development projects using technology is not trivial.

Shared context and risk of faultlines

When two or more people interact collocated, they automatically share a physical context providing rich cues such as facial expressions etc., which supports the conversation (MATTHIESEN; BJØRN, 2016; RANGANATHAN et al., 2002). A shared context can emerge, when team members share common professional language and vocabulary relevant for their work processes, work cultures, and use of digital tools potentially reducing the risk of conflicts (HINDS; MORTENSEN, 2005). However, people involved in geographically distributed projects, does not automatically can create a shared context (SCHILIT; HILBERT; TREVOR, 2002) potentially missing important contextual information, thus increasing the difficulty in identifying and solving problems, which again increase the likelihood of emerging conflicts (HINDS; MORTENSEN, 2005). Frequent interaction has been pointed to as essential for negotiation and conflict resolution (CHRISTENSEN; BJØRN, 2014; HINDS; MORTENSEN, 2005; HINDS; BAILEY, 2003). However, a high number of messages that came up into communication tools risk to unshared context once depersonalizing the interaction (SPROULL; KIESLER, 1992). While technology mediated text-based interaction generates less social presence and lack social cues compared with face-to-face conversation (POSTMES; SPEARS; LEA, 1998), the more fundamental challenge is the lack of shared context creating contextual differences. Thus, that shared context is hard to be articulated and identified during the text-based chat and consequentially cause misunderstanding among the participants (HINDS; WEISBAND, 2003). It suggests that virtual teams are likely to experience more conflict in negotiation and coordinating tasks than a

collocated team (HINDS; BAILEY, 2000). Indeed, increase social presence by establishing a shared context is relevant if we are to support technology-mediated interaction between subgroups of collocated and distributed teams. Such context-aware technology is a class of communication tools, which addresses people's knowledge context to leverage the communicative understanding (SCHILIT; HILBERT; TREVOR, 2002).

While the majority of the literature on shared context and negotiation focus on teams where all participants are geographically distributed, the situation in distributed software development is often that not every individual are geographically distributed. Instead distributed software development is often based in a situation of distributed subgroups, where several developers are collocated while subgroups are geographically distributed. When you have geographically distributed subgroups, there is a risk of faultlines. Faultlines refer to conceptual dividing lines which split a group into at least two relatively homogeneous subgroups based on group members' demographic alignment different individual attributes that impact on group processes further outcomes both performance and emotional experience (THATCHER; PATEL, 2012; BEZRUKOVA et al., 2009; SHEN; GALLIVAN; TANG, 2008). Thus, subgroup formation influences the performance of the whole group above and beyond what can be predicted by diversity alone (THATCHER; PATEL, 2012). For instance, a faultline may occur based on education level or work experience starting entirely different dynamics in a group, i.e., group members create subgroups relatively homogeneous based on informational characteristics of individuals that are directly job-related important – in this case a faultline category is information-based (THATCHER; PATEL, 2012; BEZRUKOVA et al., 2009). When team members experience problematic subgroups dynamics, it is difficult to overcome the geographically distance (CRAMTON, 2001). To create and establish task cohesion which can counter the risk of faultlines, geographically distributed teams must develop shared norms, roles, and procedures, by which they can experience accuracy of mutual comprehension (e.g., shared context). Moreover, those teams have shared expectation regarding the common goal, how to organize the interdependency and mutual trust, as well as the frequency of communication among members (LOCKWOOD; 2017, ARMSTRONG; COLE, 2002). Therefore, successful subgroup dynamics must reduce the risk of faultlines generated by time, national/regional culture, and geographical distance, in order to integrate teams from different locations providing means to sound negotiations.

Chat technology supporting negotiations

In collocated or distributed projects, communication occurs through synchronous and asynchronous means. Asynchronous communication is considered appropriated when activities have low complexity while synchronous communication is most applicable when complex activities are involved (RIOPELLE et al., 2003). However, in distributed teams frequently synchronous interactions are embedded in a broader context of asynchronous interactions and how the informal activities are carried out by the participants (OLSON; OLSON, 2000). Chat technology refers to the type of technology which allow participants to interact asynchronously through text-based interaction such as Messenger, Skype, WhatsApp, and Slack. We are currently witnessing how chat technology in

increasingly being introduced into the workplace. The usage of chat technology is thus entering the workplaces and thus becomes part of shaping the form of communication which take place in organizations. With the introduction of chat technology, we also see a decrease in the use of email, phone calls, and other means of communication (GREIF; MILLEN, 2003). In distributed software development, software developers have used chat technology in bug fixing reducing the effort of articulation work (TENÓRIO; PINTO; BJØRN, 2018), and to coordinate their activities (BODEN et al., 2014). Chat technology offers to the software developers, new advantages for their communication, since they can be modified, reviewed, and share the complete conversation over time (VAN DER ZWAARD; BANNINK, 2014). Although miscommunications cannot easily be solved when using textual interaction (FORD et al., 2017; TERUI, K.; HISHIYAMA, 2014), the ability of the chat to send short messages using informal language offers a mean for agile communication, and messages can be saved and, occasionally, retrieved and forwarded to other groups or individuals (GREIF; MILLEN, 2003). Also, the permanent nature of chat messages can form common discussion point for participants (ROBINSON, 1991). However, formal language is not always feasible, since depending upon context, i.e., a rapidly changing project environment requires an informal communication in supported by informal language use (DE VRIES; LAGO, 2010; ÅGERFALK, FITZGERALD; HOLMSTRÖM, 2005; CLERC; HERBSLEB et al., 2000). Nonetheless, both formal and informal dialogue can obstruct the conversations, if messages are shared outside the attended audience (ROBINSON; KOVALAINEN; AURAMÄKI, 2000). Professionals who share similar perspectives through professional language and knowledge makes it easier to develop common language and norms that can form a basis of communication within the distributed team (OAKLEY, 1999). This allows for healthy interactions between distributed team members facilitated by the informal language usage (HINDS; MORTENSEN, 2005). Therefore, previous researches point out how chat technology can facilitate communication at the workspace. However, our interest here is focused on how chat technology supports negotiations in geographically distributed software development team of vendors and clients.

RESEARCH METHOD

Our research is based upon an ethnographically informed (RANDALL; HARPER; ROUNCFIELD, 2007) empirical study of the software development involved in a Brazilian software company. We studied the work involved in organizing the collaborative work, focusing in on the use of technological artefacts (BLOMBERG, J.; KARASTI, 2013). In particular, we were interested in how the software development team used chat technology to support the collaboration across geographical sites of design (BJØRN; BOULUS-RØDJE, 2015). In this work, we followed a software team whose worked on a governmental IT-project: E-Account. E-Account is an information system designed to support a Brazilian municipality in organizing, monitoring, and controlling public accounts. Our interest is not the content of the E-Account project, but rather the way the software development team collaborated. In total twenty-three developers were involved in the E-Account project, we focus on how these software developers who represented both the vendor and the client negotiated using chat technology. We refer to the vendor company as BrazilSoft.

Empirical settings

The E-Account project is a Brazilian governmental IT-project in which the teams are geographically dispersed with a temporal distance of +3 hours from the vendor site to the client site. The project started in 2011 and is part of a larger information system web-Gov, which went online by the end of 2012. The web-Gov information system is designed to support the public administration of one capital in the north of Brazil, which has around 420,000 inhabitants. Currently, the web-Gov has been running live for six years and has approximately 1,200 users all municipality employees. However, the system is continuously being expanded, reconfigured, and re-designed. Thus, the web-Gov as an IT-project can be seen as ongoing infrastructure activities, which shapes how the municipality function based upon insights from the users. Furthermore, new functionalities will be made available, so the system is not only used by municipality employees but to serve 190,000 citizens in their interaction with the government.

E-account project

The E-Account is an example of a mixed operation, combining offshoring and outsourcing. While the company BrazilSoft is located in a city in the south of Brazil, the client is 3,573 kilometers to the north of the country. BrazilSoft has an offshoring operation at the client site, with a team composed of five employees, including one operation manager, one project manager, and three developers. Furthermore, there are two BrazilSoft partner-firms in outsourcing operation to support the E-Account. They are responsible for keeping the client-infrastructure and developing the web-Gov web services. The BrazilSoft local team has twenty-five employees, among them operation manager, project managers, developers, and testers. Thus, the BrazilSoft is considered a medium-sized software development company in which connects more than fifty people in the project. The communication among BrazilSoft's local team, distributed team, and the client is primarily organized in distributed settings supported by chat technology, in particular, eleven Skype chat groups and five WhatsApp groups. Each chat group has a concrete purpose and is related to a specific topic such as technical support, request changes, administration issues, contract terms, and work coordination. The client participates in some groups chat, while others are exclusive to BrazilSoft employees.

Data collection and analysis

Data were collected through interviews and observations of the interaction in the groups chat. We conducted five face-to-face interviews with vendor stakeholders (e.g., directors, project managers, and developers) during May 2017. All interviews were in Portuguese and recorded with the consent of the interviewees. During the interviews, the use of chat group technology kept appearing as critical for the negotiation practices within the team, and we decided to explore this further. In total eleven Skype chat group forums were created by BrazilSoft each aimed at interacting with clients. We obtained permission to participate as 'observer' in four Skype chat groups for four months. Thus, we were able to collect the complete interaction in the four groups chat. Our data analysis

was done in two steps. First, we listened, transcribed, and codified the interviews using ‘Express Scribe’. Second, we collected the chats scripts of the observed chat groups, which were then were transported into the Express Scribe for analyzing and codification. Both interviews and chat data were codified identifying themes in the conversation aiming to identify interesting interaction aspects. Through this process, we began to notice how the users were applying the chat technology to support their negotiations. Thus, we decided to focus on the instances of the data, where the client and the vendor were negotiating different aspects of their work such as tickets, releases, bugs, validations, and workflows. In total, we had eighteen pages of chat group transcriptions over ninety days referring to the two most active groups chat (Ticket Chat Group and BrazilSoft Private Chat Group). Table 1 gives and overview of the interaction in the two chat groups.

We observed that the chat groups had fifty-five cases where they negotiated various aspects, and in forty-three of these instances they succeed in reaching an agreement (see Table 1). All these negotiations where done using only chat technology, and no other types of technology such as email or phone were used. Thus, the interesting aspect from our perspective is that the software developers were able to negotiate successfully just only text-based chat, i.e., no video, email, or other types of technology was used. Each of the negotiations demonstrate similar patterns therefore, in the next section, we present our findings focusing on one example, to demonstrate our empirical observation.

Table 1 - Negotiations observed in the chat groups

Items	Ticket Chat Group	Private Chat Group	Total
Conversations	504	678	1182
Observed Negotiations	20	35	55
Successful Negotiations	14	29	43
Participants	11	9	20

Source: The authors

RESULTS

The web-Gov information system has been in use for six years, and the mixed-vendor/client software team in the E-Account project was created to continually identify and collect new user requirements or bugs in the system, which were to be analyzed and potentially implemented and finally be additional functionality in the production environment. The organization of the work in E-Account is ‘ticket-oriented’, which means that the coordination of activities is structured by tickets. This entails that all new tasks are organized into tickets, which are then prioritized according to the client urgencies. So, the prioritized ticket list is the main coordination tool for the software developers. In order to organize the work, all new user requirements are included into the software management repository called Redmine. Redmine is a web-based open-source software management application designed to coordinate requirements. Thus, each requirement is created as ‘tickets’ into the Redmine repository. The client (the municipality) is responsible for accessing and creating tickets in the Redmine, including describing

each requirement and defining its priority. BrazilSoft's developers then access the Redmine to identify requirements, while assigning themselves as responsible for particular tickets. The BrazilSoft project manager also access the Redmine on a regular basis monitoring the status of all tickets. When a ticket is done, the developer record this in the Redmine, and the client validates the ticket, and, if approved, inform BrazilSoft developers to include the ticket into the web-Gov production environment. However, over the last years, the ticket quantity has increased considerably, and BrazilSoft have experienced several clients claims regarding delays to include validated tickets into the production environment.

Despite the ticket control embedded in the Redmine, the ticket-oriented process was continually failing, since the client frequently forgot to validate tickets or the vendor forgetting to include them in the production environment. These events increased customer's complaints regarding unavailable features in the web-Gov system and increased the tension in the vendor-client relationship. Attempting to avoid client claims, 18 months before our research, the vendor introduced chat group technology in order to monitor the ticket-oriented process. The vendor intention was to streamline the coordination of the tickets. Concretely, the vendor notifies the client of the tickets, which require validation before including them in the production environment. The 'ticket chat group' was successful in the first three months, however issues began to arise. Communication breakdowns took the form of the client forgetting to report in the 'ticket chat group' which tickets were validated and thus ready for inclusion in the web-Gov production environment. Delays became a large problem, and due to the contractual structure between the vendor and client, delayed tickets would mean that the vendor had to pay fines to the client. The increase numbers of fines in the project became a stress-point for the client-vendor relationship and generating conflicts among the cross-organizational team. The conflict was openly available to everybody, since it took place in the 'ticket chat group', exposing the problems to all participants. We observed forty-seven messages exchanged in the chat group concerning the issues of delayed validations and fines. Below we zoom in of the core exchanges. Following quotation exemplify the issues between the client and John, the project manager at the vendor site.

Client: "How come that ticket [ID-number] isn't yet in the production environment."

John (BrazilSoft site): "Because the ticket has not been validated by you yet."

Client: "Did you ask me [to validate the ticket] through notification features in the ticket chat group?"

John (BrazilSoft site): "No, I forgot, sorry. But you could look at Redmine. See the ticket [a picture was posted in the chat group]. What you see here is that there is a red alert [see the screen shot]. This red alert means, we are waiting for you to validate the ticket before we can proceed."

Client: "This practice is not what we agreed on. We decided that our work routine for validation of ticket by us but go through the ticket chat group. You and the others MUST notify me in the request

forum.”. Ticket chat group, via Skype, Jun 10th, 2017 (translated from Portuguese)

A few days later, after the discussion above, Peter, a project manager at the client site, suggested internally at a face-to-face meeting with the client to replace the current ticket-oriented process with a release-oriented process. Peter argues that the adoption of the release-oriented process would ‘pack’ a set of tickets into one release and it would facilitate its validation. Consequently, the messages exchanged in the ‘ticket chat group’ regarding ticket validation would also be reduced once a release contains a set of tickets rather than individual tickets. Potentially, conflicts concerning ticket would be avoided. However, such a change would require quite some differences to the way the work was organized, but contractually but also processes oriented. Thus, a longer negotiation concerns the possibility to change the process was initiated. This negotiation took place in two chat groups, and it all began in the ‘ticket chat group’.

Peter (client site): “Hi guys. Yesterday, I had a meeting with the infrastructure team and [client name], where we discussed our workflow. Thus, who will decide what to include in the production environment. The decision is ultimately the system manager [client name]. However, I suggested to update our current work routine replacing ticket-oriented process by a release-oriented process. They liked the idea however, we need to discuss this idea, and how to proceed.”. Ticket chat group, via Skype, Jun 22th, 2017 (translated from Portuguese)

What happens in the above quotation is that Peter, one of the core software developers, whose is located at the client site explains, how he has been discussing a potential new way of organizing the workflow in the team. More importantly, he also suggests concrete changes and supports it by referring to that the client has approved of the idea. Consequentially, a team member from the client site also writes a message in the chat group, supporting Peters idea, further demonstrating that the client supports the idea.

Client: “Hello, everyone. As [name of the project manager at client site] wrote, we are excited to adopt the release-oriented process. As far as I know, this will make our validation process much easier. We are looking forward to adopting it.”. Ticket chat group, via Skype, Jun 22th, 2017 (translated from Portuguese)

So, we have a situation, where people collocated in the project have had important discussions about the workflow, and moreover, it is important to notice that while both above participants are collocated at the client geographical locations. Indeed, they represent two different parties, namely the client and the vendor. Meanwhile, at the vendor’s geographical location, the idea for change is not fully embraced. However, to have such a discussion internally within the vendor team, before including the client, the project manager created a new discussion forum in ‘BrazilSoft private chat group’ in which John, the software developer in BrazilSoft, but working remotely from the client, resisted the idea of changing the workflow towards release-oriented process.

John (BrazilSoft site): “I’m tough about this situation [replacing ticket-oriented by release-oriented] because we risk increasing our delay

since the client then want to include additional new functionalities for each production release. Currently, the client already delayed their validation of new functionalities, so if we adopt this new process, the delay could increase because they will wait to include a new set of functionalities together in the production environment. Maybe it does not avoid the complaints about the existing delay.”. BrazilSoft private chat group, via Skype, Jun 22th, 2017 (translated from Portuguese)

What is interesting here, is that the discussion moved from the ‘ticket chat group’ to the ‘BrazilSoft private chat group’. The private chat forum allowed the vendors to negotiate within BrazilSoft excluding the vendor however still including all BrazilSoft employees also the ones located at the vendor sites. The negotiation continues, and Peter attempts to convince John that the opportunity to move towards a release-oriented process is appropriate and supporting the software developers in BrazilSoft.

Peter (client site): “I agree with you, but if we adopt release-oriented process, everything that is done within the release goes to the production together in a short time. Moreover, we always wanted to adopt release-oriented process. It is great chance for us.”. BrazilSoft private chat group, via Skype, Jun 22th, 2017 (translated from Portuguese)

The synchronous interaction continues and John resists Peter’s argument to adopt the release-oriented process. He refers to the timing of the change and how it might drastically change their current workflow causing problems. John explains how such a change is not trivial, but instead involves complex changes to their existing workflow review processes.

John (BrazilSoft site): “I agree, but I think we shouldn’t do this now. I think that is a bad idea because it changes drastically our current work routine which demands a review of our work flow.”. BrazilSoft private chat group, via Skype, Jun 22th, 2017(translated from Portuguese)

Following this interaction, multiple different opinions and concerns are presented in the chat group. Evidently, the interaction leads to a conflict within the vendor team between the project managers John and Peter (both working for BrazilSoft, however geographically located at different sites). The main issue is the impact which the potential change will have on their workflow review process. Trying to resolve the issue, BrazilSoft’s operation manager enter the negotiation.

Operation Manager (BrazilSoft site): “Hey guys. Currently, they do this! I think it doesn’t have a significant impact on our current work flow. Just few adjustments.”. BrazilSoft private chat group, via Skype, Jun 22th, 2017 (translated from Portuguese)

The operation manager tried to make the issue less controversial. Moreover, another vendor software developer also enters the discussion supporting both the operation manager and argue to make the change in the release process. In this negotiation, it is important that the chat technology allow people to enter the negotiation over time, and thus the ‘BrazilSoft private chat group’ provides a shared context supporting discussion and negotiation across the geographically

dispersed developers. Thus, we now have a case, where people on both geographical sites agree and support the change. However, it is important to notice that John (who still resist the change) is a core employee and his opinion matters, even though other developers approve of the change, as shown below.

Developer (BrazilSoft site): “They'll continue doing what they always do. I don't think that is a problem to adopt release-oriented now. It'll facilitate our work reducing the current validation problems.”. BrazilSoft private group, via Skype, Jun 22th, 2017 (translated from Portuguese)

Furthermore, the operation chat manager also decides to modify his first opinion to support John, by saying that once they have made this change, it will be impossible to return to the previous workflow. Thus, they should be entirely sure that it is a good idea to replace the ticket-oriented process with the release-oriented workflow.

Operation Manager (BrazilSoft site): “What they need to understand is that once adopted release-oriented there is no how to get back. I mean, everything in it must be validated as release. [...] It is because there is no way to separate the codes after being integrated. That would improve our work routine.”. BrazilSoft private chat group, via Skype, Jun 22th, 2017 (translated from Portuguese)

Peter, who was the one starting the whole discussion then copy and paste the message from the client, which originally was posted in the other skype group, namely the ‘ticket chat group’. He follows the pastes message by arguing that the ticket-oriented workflow process is currently not working. The issue is that client frequently forgets, which tickets must be validated, thus loses control over the process and everybody gets delayed.

Peter (client site): “I reinforce that a ticket-oriented is not is not good for us because them [client] is not validating each ticket due the high-ticket quantity. Thus, they are forgetting to validate each ticket due it is hard to control. This is the reason why the release-oriented process can figure it out. I'd also like to highlight that at the client meeting, yesterday, everyone [client names] agreed with this change commenting that it can be good for all of us. In addition, in our ticket chat group [client name] wrote: ‘[message pasted from ticket chat group]’. Then, we shouldn't lose this opportunity to change and improve our process.”. BrazilSoft private chat group, via Skype, Jun 22th, 2017 (translated from Portuguese)

The local project manager then agrees to the new process and his colleagues to adopt release-oriented delivery. Nonetheless, he suggested that a workflow process was designed, presented, and approved by all aiming to make clear the new process to the client.

John (BrazilSoft site): “Okay, I agree only if we design a workflow formalizing this process [release-oriented. And they [client] need to approve the workflow proposed by us. The workflow will be our guarantee of the agreement. I can design and send the workflow to them.”

Operation Manager (BrazilSoft site): “I agree.”

Peter (client site): “OK. [emoticon with smile]”. BrazilSoft private chat group, via Skype, Jun 22th, 2017 (translated from Portuguese)

The project manager at client site sent a ‘like’ sign in the group in which he agreed with the idea. The next day, the negotiation of the release-oriented versus ticket-oriented work structure moved from the private chat group to ‘public’ ticket chat group in which the project manager at vendor site sent a message accepting the change.

John (BrazilSoft site): “Ok [client] we agree, and we’ll design the release-oriented process in a workflow to be approved for you all. I’ll send the workflow soon.”

Client: “Good news! I’m looking forward to seeing the workflow.”

Peter (client site): “(Y) [Thumb up emoticon]”. Ticket chat group, via Skype, Jun 23th, 2017 (translated from Portuguese)

On July 14th, 2017, John shares a document describing the first version of the workflow in BrazilSoft’s private chat group and invites participants to validate it. The operation manager and the administrative manager (whose also participates the group) suggested few adjustments. John sent a second version two hours later, which is approved by all participants in BrazilSoft’s private chat group. Afterwards, the local project manager sent the final version of the workflow in the inclusive ‘ticket chat group’, where the client approved it a few hours later.

Analyzing the negotiation process in the chat groups, we observed that the discussion moved dynamically between the subgroups – from the restricted ‘ticket chat group’ to the inclusive ‘BrazilSoft’s private chat group’. While the negotiations might on the surface seem as a discussion on the work process, it was, in fact, also a demonstration of a power struggling between the two geographically distributed project managers. Once Peter works at the client site on a daily basis, he also took the liberty to suggest a workflow change without consulting with John’s working at the vendor site. Thus, when John first learn that Peter has made a proposal to the client on a drastic change for the client-vendor relationship, without consulting him. So, John became critical and the potential conflict began to arise, which was initiated in the ‘public’ chat group but moved into the private’s vendor chat group. Moreover, even though chat technology is an asynchronous interaction, we perceive in all the messages in the above examples almost went as a synchronous interaction. So, while chat technology fundamentally is an asynchronous technology, it allowed Peter and John to negotiate the workflow changes promptly with the participation of other colleagues, who give their opinions voluntarily. What made the negotiation successful was that the chat technology enabled the participants in both formal and information language exchanges, where they constantly could move between levels of negotiations. In this way, the double-language level (i.e., informal and formal language) allowed the participants to develop a shared context which supported multiple people in navigating across subgroup, language levels, utilizing the permanent record created by the chat technology.

DISCUSSION

From the software vendor perspective, the issue about the workflow change produced a delicate situation. Concretely, Peter, the project manager at the client site, had initiated an unauthorized negotiation with the client, without first checking the vendor's opinion. Thus, by initiating the negotiation with the client, Peter also impacted the client expectation towards the vendors' interest in moving towards release-oriented processes. This situation meant that it was important for Peter to convince John that the release-oriented process was the way to go, since if he failed, he would have to face the client and explain how this process change was not good with the risk of losing face.

Luckily for Peter, the software development team manage to successfully negotiate and solve their challenges concerning how to organize the collaborative process, despite being geographically distributed interacting using the online chat group. Prior research has pointed out to how negotiation and miscommunications cannot easily be solved using primarily textual interaction (FORD et al., 2017; TERUI, K.; HISHIYAMA, 2014; BJØRN; HERTZUM, 2006; VALLEY; MOAG; BAZERMAN, 1998), due to the lack of implicit clues and spatial references, which supports the creation of a shared context (HINDS; BAILEY, 2000; SPROULL; KIESLER, 1992). When people are collocated, they are able to use gestures and facial expressions to indicate through feedback loops how they are interpreting the situation supporting negotiations. In this sense, the question becomes *what made the negotiation a success despite the lack of feedback and contextual information? How did the chat group technology allow for the distributed software developers to reach an agreement?* Our data extend prior CSCW research on negotiation protocols for work (ESBENSEN; BJØRN, 2014) and the use of chat technology in organizations (RIEMER; FRÖSSLER; KLEIN, 2007) in several ways.

Firstly, our data show that the textual and permanent nature of the chat group technology was crucial for supporting the negotiation between the vendor and the client. Prior research on chat technology (JOWETT, 2015; LI; ROSSON, 2014; HSIUNG, 2000, IM; CHEE, 2006) also supports this finding, since they point out that keeping conversation history is an essential feature in chat technology. By saving the complete conversation history it is possible for users to access and analyze prior conversations (VAN DER ZWAARD; BANNINK, 2014) supporting reflective behavior and potential re-submission of past interaction in new conversations. While participants might choose to exclude or delete past messages in certain chat interactions, such action will be registered in the conversation history and made visible available to all participants in the chat group. Analyzing our data, we observed how the vendor's private chat group made use of the permanent records by copying and pasting previous client-messages from the other chat forum to reinforce argumentation. The permanent record was not only used as a way to review the past interaction, but also to document past behavior facilitating a shared context supporting the negotiations – as in pointing out explicitly what the object of concern entails. By pasting in quotations from earlier, the participants were able to 'gesture' and 'pointing' towards the area of concern, thus supporting grounding activities (CLARK; BRENNAN, 1991; SEGENREICH, 2008) in the conversation.

Secondly, we found that the synchronous interaction embedded in chat technology supported the negotiation. The participants pointed out that

navigating substantial email conversations is often problematic, and it is difficult to comprehend and follow the different lines of interaction fully. Furthermore, prior work has demonstrated how email technology lack feedback to the sender from the receiver increasing risk of misunderstandings (BJØRN; NGWENYAMA, 2009). For instance, it is not possible to know whether their emails have been seen by others and also whether they are, actually, doing something about it. In the chat technology, you can see whether others have seen the messages and identify who is available, and even more importantly – others can monitor the interaction of others, without interfering directly (TENÓRIO; PINTO; BJØRN, 2018). Thus, the ways in which the chat technology through the permanent features, the informal language, and then also supporting the reviewability of others to monitor the interaction in ‘synchronous’ way of others facilitated the successful negotiation.

Thirdly, our data shows that the chat technology made it possible for the participants to interact informally, compared to their otherwise formal textual interactions in their email use. While the permanent feature of emails requires participants to interact using formal language to ensure accurate interpretation, the permanent features of the chat technology were very different. In the chat technology, participants were allowed to informally interact developing a cultural language (i.e., double-language level) of interaction and interpretation (ROBINSON, 1991), in which ‘items’ of concern were transformed from formal interpretation to a common understanding (OAKLEY, 1999; ROBINSON, 1991). This was evident in the situations, where we saw how the participants did not spend any time nor effort on using formal contextual language in their messages. Instead, participants jump right into the issues of concerns. While the formal communication (e.g., emails) is driven by the highly specific context (LOCKWOOD, 2017), the chat interaction facilitates informal interaction. Thus, chat technology supported the participants in grounding activities in the negotiation. During our interviews, participants mentioned several times that they perceived the chat technology to be fast, which was related to the informal language supporting ‘direct talk’ (HINDS; MORTENSEN, 2005; ROBINSON, 1991). In addition, the participants considered it comfortable to use the chat technology, since it allowed them ‘to query one’s entire team at once’ (HERBSLEB et al., 2002).

Finally, we found that chat technology help to reduce the risk of subgroup dynamics causing faultlines (CRAMTON; HINDS, 2004). When teams are composed of geographically distributed subgroups, where demographic attribute align, there is a risk of creating faultlines complicating collaboration. A risk which is further strengthen in cases, where other types of distinct features confirm the differences across sites, such as like nationality or seniority (MATTHIESEN; BJØRN, 2016). Chat technology made it possible for participants to divide their interaction into parallel groups of interactions, which each created and shaped subgroups in different ways – both across and within geographical locations. In our case, the participants divided their interaction into two main chat groups: the inclusive ‘ticket chat group’ and the excluding ‘BrazilSoft’s private chat group’. However, by having these pre-defined forums, with existing pre-defined participants and purposes, users did not have to consider who to send potential information to each time they were sending a message. They did not risk forgetting to add others or include the wrong audiences for their messages. Instead, the pre-determined nature of participation made it possible for participants to utilize the permanent nature, the informal language, and the reviewability and navigation of the conversations in a fast and informal way, making the negotiation similar as to if the participants had

been collocated. In this way, the chat technology allowed the participants to navigate and organize subgroups, while supporting collaboration across subgroups, thus reducing the risk of faultlines. Table 2 summarizes our relevant findings which supported a successful negotiation within a Brazilian distributed software development team. Those findings answer our research question once chat technology enables textual and permanent nature of the conversation, the synchronous interaction embedded, informal interaction among the participants, and, finally, to reduce the risk of faultlines. Therefore, our findings can drive new the design of cooperative technologies supporting geographically distributed collaboration.

Table 2 - Chat technology supporting successful negotiation

Findings	Description
Textual and permanent nature	Textual and permanent nature of the chat technology was crucial for supporting the negotiation between the vendor and the client.
Synchronous interaction	The synchronous interaction embedded in chat technology supported the negotiation.
Informal language	Chat technology made it possible for the participants informally interaction when compared to their otherwise regular textual interactions, in particular, email.
Reduce faultlines	Chat technology helps to reduce the risk of subgroup dynamics causing faultlines.

Source: The authors

CONCLUSION

This study investigated a successful negotiation within a Brazilian distributed software development team using chat technology. We found that the chat technology facilitated negotiation by providing a shared context, synchronous interaction embedded in asynchronous functionality, combined with reviewability supporting navigation by the participants. Analyzing the two chat groups and interviewing their participants, we observed that permanent nature, informal language, navigation, and pre-defined features of subgroups were salient for the success of the negotiation and resolving a potential critical conflict between two core software developers who were geographically distributed. We argue that chat technology has clear strengths in terms of supporting critical interaction within organizations, thus, when we, as CSCW researchers, are to explore and design cooperative technologies supporting geographically distributed collaboration. Therefore, we consider the feature of chat technology and how such features can be generally embedded into the multiple cooperative technologies supporting distributed collaboration both within and outside of the software development domain.

ACKNOWLEDGMENT

We would like to thank Cesumar Institute of Science, Technology, and Innovation (*Instituto Cesumar de Ciência, Tecnologia e Inovação – ICETI*), Maringá, Paraná, Brazil and MGA Public Management (*MGA Gestão Pública Ltda.*), Maringá, Paraná, Brazil.

REFERENCES

ÅGERFALK, P. J.; FITZGERALD, B.; HOLMSTRÖM, H.; et al. A framework for considering opportunities and threats in distributed software development. Proc. International Workshop on Distributed Software Development, Paris, France: Austrian Computer Society, n. August, p. 47–61, 2005.

ANDERSON, T.; KANUKA, H. On-Line Forums: New Platforms for Professional Development and Group Collaboration. *Journal of Computer-Mediated Communication*, v. 3, n. 3. Blackwell Publishing Ltd., 1997.

ARMSTRONG, D. J.; COLE, P. Managing distances and differences in geographically distributed work groups. In: P. Hinds; S. Kiesler (Orgs.); *Distributed work*. 1o ed, p.167–186. Cambridge, MA: MIT press, 2002.

BEZRUKOVA, K.; JEHN, K. A.; ZANUTTO, E. L.; THATCHER, S. M. B. Do Workgroup Faultlines Help or Hurt? A Moderated Model of Faultlines, Team Identification, and Group Performance. *Organization Science*, v. 20, n. 1, p. 35–50, 2009.

BJØRN, P.; BARDRAM, J.; AVRAM, G.; BANNON, L; BODEN, A.; REDMILES, D., DE SOUZA, C.; WULF, V. Global software development in a CSCW perspective. Proceedings of the companion publication of the 17th ACM conference on Computer supported cooperative work & social computing - CSCW Companion '14. *Anais...* . p. 301–304. New York, New York, USA: ACM Press, 2014.

BJØRN, P.; BOULUS-RØDJE, N. The multiple intersecting sites of design in CSCW research. *Computer Supported Cooperative Work (CSCW)*, v. 24, n. 4, p. 319–351. Springer, 2015.

BJØRN, P.; ESBENSEN, M.; JENSEN, R. E.; MATTHIESEN, S. Does Distance Still Matter? Revisiting the CSCW Fundamentals on

Distributed Collaboration. *ACM Transactions on Computer-Human Interaction*, v. 21, n. 5, p. 1–26, 2014.

BJØRN, P.; HERTZUM, M. Proactive behaviour may lead to failure in virtual project-based collaborative learning. *Proceedings of the 2005 international ACM ...*, p. 326–327, 2005.

_____. Project-Based Collaborative Learning: Negotiating Leadership and Commitment in Virtual Teams. *5th Conference on Human Computer Interaction in Southern Africa (CHI-SA)*. Anais... . p.6–15. Cape Town, South Africa: ACM SIGCHI, 2006.

BJØRN, P.; NGWENYAMA, O. Virtual team collaboration: Building shared meaning, resolving breakdowns and creating translucence. *Information Systems Journal*, v. 19, n. 3, p. 227–253, 2009.

BLOMBERG, J.; KARASTI, H. Reflections on 25 years of ethnography in CSCW. *Computer Supported Cooperative Work (CSCW)*, v. 22, n. 4–6, p. 373–423, 2013. Springer.

BODEN, A.; ROSSWOG, F.; STEVENS, G.; WULF, V. Articulation spaces: Bridging the Gap between Formal and Informal Coordination. *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing - CSCW '14*. Anais... . p.1120–1130. New York, New York, USA: ACM Press, 2014.

CHRISTENSEN, L. R.; BJØRN, P. Documentscape: Intertextuality, Sequentiality, & Autonomy at Work. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*. Anais... . p.2451–2460. New York, New York, USA: ACM Press, 2014.

CLARK, H. H.; BRENNAN, S. E. Grounding in communication. In: *L. Resnick; J. Levine; S. Teasley (Orgs.); Perspectives on socially shared cognition*. p.127–149. Washington: American Psychological Association, 1991.

CLÉMENT, R.; BAKER, S. C.; MACINTYRE, P. D. Willingness to Communicate: can online chat help? *International Journal of Applied Linguistics*, v. 16, n. 2, p. 189–212, 2003.

CLERC, V.; DE VRIES, E.; LAGO, P. Using wikis to support architectural knowledge management in global software development. Proceedings of the 2010 ICSE Workshop on Sharing and Reusing Architectural Knowledge - SHARK '10. Anais... . p.37–43. New York, New York, USA: ACM Press, 2010.

CRAMTON, C. D. The Mutual Knowledge Problem and Its Consequences for Dispersed Collaboration. *Organization Science*, v. 12, n. 3, p. 346–371, 2001.

CRAMTON, D. C.; HINDS, P. J. Subgroup dynamics in internationally distributed teams: Ethnocentrism or cross-national learning? *Research in Organizational Behavior*, v. 26, n. 04, p. 231–263, 2004.

DABBISH, L. A.; KRAUT, R. E.; FUSSELL, S.; KIESLER, S. Understanding Email Use: Predicting Action on a Message. Proceedings of the 2005 Conference on Human Factors in Computing Systems (CHI), p. 691–700, 2005.

ESBENSEN, M.; BJØRN, P. Routine and standardization in global software development. Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work, p. 12–23, 2014.

FAYARD, A.-L.; DESANCTIS, G. Evolution of an Online Forum for Knowledge Management Professionals: A Language Game Analysis. *Journal of Computer-Mediated Communication*, v. 10, n. 4. Blackwell Publishing Ltd., 2005.

FORD, C.; GARDNER, D.; HORGAN, L. E.; LIU, C.; NARDI, B.; RICKMAN, J. Chat Speed OP PogChamp: Practices of Coherence in Massive Twitch Chat. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2017), p. 858–871, 2017.

GREIF, I.; MILLEN, D. R. Communication Trends and the On-Demand Organization. IBM - T.J. Watson Research Center - Cambridge, MA, 2003.

GUO, ZI.; D'AMBRA, J.; TURNER, T.; ZHANG, H. Improving the Effectiveness of Virtual Teams: A Comparison of Video-Conferencing and Face-to-Face Communication in China. *IEEE*

TRANSACTIONS ON PROFESSIONAL
COMMUNICATION TRANSACTIONS ON PROFESSIONAL
COMMUNICATION, v. 52, n. 1, p. 1–16, 2009.

HERBSLEB, J. D. Global Software Engineering: The Future of Socio-technical Coordination. Future of Software Engineering (FOSE '07). Anais... . p.188–198. IEEE, 2007.

HERBSLEB, J. D.; ATKINS, D. L.; BOYER, D. G.; HANDEL, M.; FINHOLT, T. A. Introducing Instant Messaging and Chat in the Workplace. Proceedings of the Conference on Human Factors in Computing Systems (CHI'02), n. 4, p. 171–178, 2002.

HERBSLEB, J. D.; MOCKUS, A.; FINHOLT, T. A.; GRINTER, R. E. Distance, dependencies, and delay in a global collaboration. Proceedings of the 2000 ACM conference on Computer supported cooperative work - CSCW '00. Anais... . p.319–328 New York, New York, USA: ACM Press, 2000.

HERBSLEB, J. D.; PAULISH, D. J.; BASS, M. Global software development at siemens: experience from nine projects. Proceedings. 27th International Conference on Software Engineering, 2005. ICSE 2005. Anais... . p.524–533. IEEE, 2005.

HINDS, P. J.; BAILEY, D. E. Virtual Teams: Anticipating the Impact of Virtuality on Team Process and Performance. Academy of Management Proceedings, v. 2000, n. 1, p. C1–C6, 2000.

_____. Out of Sight, Out of Sync: Understanding Conflict in Distributed Teams. Organization Science, v. 14, n. 6, p. 615–632, 2003.

HINDS, P. J.; MORTENSEN, M. Understanding Conflict in Geographically Distributed Teams: The Moderating Effects of Shared Identity, Shared Context, and Spontaneous Communication. Organization Science, v. 16, n. 3, p. 290–307, 2005.

HINDS, P. J.; WEISBAND, S. P. Knowledge sharing and shared understanding in virtual teams. In: C. B. Gibson; S. G. Cohen (Orgs.); Virtual teams that work: Creating conditions for virtual team effectiveness. 1o ed, p.21–36. San Francisco, CA: Jossey-Bass, 2003.

HINDS, P.; RETELNY, D.; CRAMTON, C. In the Flow, Being Heard, and Having Opportunities: Sources of Power and Power Dynamics in Global Teams. *Computer Supported Cooperative Work (CSCW)*, p. 864–875, 2015.

HSIUNG, R. C. The Best of Both Worlds: An Online Self-Help Group Hosted by a Mental Health Professional. *CyberPsychology & Behavior*, v. 3, n. 6, p. 935–950, 2000.

IM, E. O.; CHEE, W. An Online Forum as a Qualitative Research Method. *Nursing Research*, v. 55, n. 4, p. 267–273, 2006.

JENSEN, R. E. Why closely coupled work matters in global software development. *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work*, p. 24–34, 2014.

JOWETT, A. A Case for Using Online Discussion Forums in Critical Psychological Research. *Qualitative Research in Psychology*, v. 12, n. 3, p. 287–297, 2015.

LI, N.; ROSSON, M. B. Using annotations in online group chats. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 863–866, 2014.

LOCKWOOD, J. An analysis of web-chat in an outsourced customer service account in the Philippines. *English for Specific Purposes*, v. 47, p. 26–39, 2017.

MALHOTRA, A.; MAJCHRZAK, A.; CARMAN, R.; LOTT, V. Radical Innovation Without Collocation: A Case Study at Boeing-Rocketdyne. *Mis Quarterly*, v. 25, n. 2, p. 145, 2001.

MARK, G.; GONZALEZ, V.; SARINI, M.; SIMONE, C. Supporting articulation with the reconciler. *CHI '02 extended abstracts on Human factors in computing systems - CHI '02*, p. 814, 2002.

MATTHIESEN, S.; BJØRN, P. Let's look outside the office: Analytical lens unpacking collaborative relationships in global work. *COOP 2016: Proceedings of the 12th International Conference on the Design of Cooperative Systems*, 23-27 May 2016, Trento, Italy. *Anais...* . p.107–122, 2016.

MORAES, G. H. S. M. de; CABELLO, O. G. The information technology and the university: use of educational applications by students. *R. Tecnol. Soc.*, Curitiba, v. 13, n. 28, p. 55-71, mai./ago. 2017.

OAKLEY, J. G. Leadership Processes in Virtual Teams and Organizations. *Journal of Leadership Studies*, v. 5, n. 3, p. 3–17, 1999.

OLSON, G. M.; OLSON, J. S. Distance matters. *Human-Computer Interaction*, v. 15, n. 2, p. 139–178, 2000.

POSTMES, T.; SPEARS, R.; LEA, M. Breaching or building social boundaries? SIDE-effects of computer-mediated communication. *Communication research*, v. 25, n. 6, p. 689–715. Sage Publications: London, 1998.

RANDALL, D.; HARPER, R.; ROUNCEFIELD, M. *Fieldwork for Design*. London: Springer London, 2007.

RANGANATHAN, A.; CAMPBELL, R. H.; RAVI, A.; MAHAJAN, A. ConChat: a context-aware chat program. *IEEE Pervasive Computing*, v. 1, n. 3, p. 51–57, 2002.

RIEMER, K.; FRÖSSLER, F.; KLEIN, S. Real Time Communication - Modes of Use in Distributed Teams. 15th European Conference on Information Systems, St. Gallen (CH), 07- 09 June, p. 286–297, 2007.

RIOPELLE, K.; GLUESING, J. C.; ALCORDO, T. C.; BABA, M.; BRITT, D.; MCKETHER, W.; MONPLAISIR, L.; RATNER, H.H.; WAGNER, K.H. Context, task, and the evolution of technology use in global virtual teams. *Virtual teams that work: Creating conditions for virtual team effectiveness*, p. 239–264. Jossey-Bass San Francisco, 2003.

ROBINSON, M. Double-Level Languages and Co-operative Working Introduction. *AI & Society*, n. 5, p. 34–60, 1991.

ROBINSON, M.; KOVALAINEN, M.; AURAMÄKI, E. Diary as dialogue in papermill process control. *Communications of the ACM*, v. 43, n. 1, p. 65–70, 2000.

SCHILIT, B. N.; HILBERT, D. M.; TREVOR, J. Context-aware communication. *IEEE Wireless Communications*, v. 9, n. 5, p. 46–54, 2002.

SEGENREICH; S. C. D. Tecnologia na avaliação da aprendizagem colaborativa on line: contribuição do fórum de discussão. *Revista Tecnologia e Sociedade* v. 4, n. 6, 2008.

SHEN, Y.; GALLIVAN, M.; TANG, X. The Influence of Subgroup Dynamics on Knowledge Coordination in Distributed Teams: A Transactive Memory System and Group Faultline Perspective. *Twenty Ninth International Conference on Information Systems, Paris 2008 Proceedings. Anais... . p.1–18. Paris: Association for Information Systems (AIS) Electronic Library (AISeL), 2008.*

ŠMITE, D.; MOE, N. B.; ÅGERFALK, P. J. *Agility across time and space: Implementing agile methods in global software projects.* Berlin, Germany: Springer, 2010.

SØDERBERG, A. M.; KRISHNA, S.; BJØRN, P. Global software development: Commitment, trust and cultural sensitivity in strategic partnerships. *Journal of International Management*, v. 19, n. 4, p. 347–361, 2013.

SPROULL, L.; KIESLER, S. *Connections: New ways of working in the networked organization.* Boston, MA: MIT press, 1992.

TENÓRIO, N.; PINTO, D.; BJØRN, P. Accountability in Brazilian Governmental Software Project: How Chat Technology enables Social Translucence in Bug Report Activities. *Computer Supported Cooperative Work (CSCW)*, p. 1-26. June 11th, 2018.

TERUI, K.; HISHIYAMA, R. Cross-cultural communication protocol analysis. *Proceedings of the 5th ACM international conference on Collaboration across boundaries: culture, distance & technology - CABS '14*, p. 95–98, 2014.

THATCHER, S. M. B.; PATEL, P. C. Group Faultlines: A Review, Integration, and Guide to Future Research. *Journal of Management*, v. 38, n. 4, p. 969–1009, 2012.

VALLEY, K. L.; MOAG, J.; BAZERMAN, M. H. "A matter of trust": Effects of communication on the efficiency and distribution of outcomes. *Journal of Economic Behavior and Organization*, v. 34, n. 2, p. 211–238, 1998.

VAN DER ZWAARD, R.; BANNINK, A. Video call or chat? Negotiation of meaning and issues of face in telecollaboration. *System*, v. 44, n. 1, p. 137–148. Elsevier Ltd., 2014.

Recebido: 06 ago 2018.

Aprovado: 16 jan 2019.

DOI: [10.3895/rts.v15n37.8655](https://doi.org/10.3895/rts.v15n37.8655)

Como citar: TENÓRIO, N. e BJORN, P., How a Geographically Distributed Software Team Managed to Negotiate Successfully using Chat Technology. *R. Technol. Soc.*, Curitiba, v. 15, n. 37, p. 164-182, jul/set. 2019. Disponível em: <<https://periodicos.utfpr.edu.br/rts/article/view/8655>>. Acesso em: XXX.

Correspondência:

Direito autoral: Este artigo está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.

