How Social Software Supports Communicative and Coordinative Practices in Global Software Development

Rosalba Giuffrida

Software and System Section

IT University of Copenhagen

A thesis submitted for the degree of

Doctor of Philosophy

August 2014

To my sweet family

Acknowledgements

I would like to express my sincere and deep gratitude to my supervisor Dr. Yvonne Dittrich for her excellent guidance through my doctorate. She has been a critical and very encouraging supervisor and she continuously supported me during the last four years.

My gratitude goes also to the members of my doctoral committee Dr. Irina Shklovski, Dr. Jacob Nørbjerg and Prof. Walt Scacchi for their insightful comments and suggestions.

I am very grateful to Dr. Jasper Overgaard, the DHI team and all the students that participated to the fieldwork I conducted during my research: without their help this work would not have been possible.

Furthermore, I would like to thank all the colleagues that I had the honour and pleasure to collaborate with during my studies at IT University of Copenhagen and all the friends I met: Christina, Naveen, Archana, Hugo, Andrea P., Gian, Wusheng, Helge, Kosta, Dario, Andrea C., Maria, Josu, Maxime, Jasmine, Stefan, Freja, Pernille, Julia, Yohan, Ewa, Nanna, Signe, Marie, Johan, Marco, Hina, Helen, Laura, Ornela.

A special "thank you" goes to Paolo Tell for the wonderful friendship and continuous support. For all the moments spent discussing research topics, sharing PhD frustrations and enjoying never-ending evening chatting and playing board games.

A big "thank you" to Maria and Nuccio, Dino and Anita, Francesco and Elena: this thesis is also the product of your support, your patience and your love.

Finally, infinite gratitude to Paolo for being an adorable husband, a sweet father, a clever researcher and an inseparable life companion, without whom I would not be here.

Abstract

Global Software Development (GSD) is increasingly becoming a common practice in the software industry. In GSD, the lack of face-to-face communication is a major challenge, and effective computer-mediated practices are necessary. Communication through Social Software (SoSo) seems to be able to support team coordination and help to deal with geographical distance; however, research in the field is at an early stage. This PhD thesis investigates the use of SoSo in three GSD projects, highlighting the usage of SoSo to overcome the lack of face-to-face communication and the role of SoSo within the ecology of channels used by the teams.

Three ethnographically-inspired studies were conducted during this PhD project. Data collection techniques are similar in the different projects, including direct observations, interviews, and document analysis, while data analysis was performed using different conceptual tools in the different phases of this PhD project. In particular, from the analysis of the field material performed in the initial part of this PhD, the necessity to describe observed practices with appropriate conceptual tools emerged. Thus, a conceptual framework based on the concepts of coordination mechanisms and communicative genres was developed in the final part of this PhD, not only to analyze and describe the role of SoSo in a GSD project, but also to show how coordinative and communicative practices are established and maintained in GSD teams.

This PhD thesis highlights the role of SoSo as a flexible tool that can be used by GSD teams as a side channel to complement collaborative Software Engineering tools that provide templates for coordination mechanisms. SoSo can support different communicative genres, including work discussions, knowledge sharing, team building and articulation work. SoSo appears to be suitable to establish, develop, and maintain social protocols during the collaboration, assuming a central role in the ecology of channels used by team members. SoSo enables team building chats and metawork, both of which are necessary for establishing successful collaboration that deals with geographical distance. Providing

distributed teams the access to flexible tools, such as SoSo, and allowing social talk through SoSo influence the effective usage of coordination mechanisms, resulting in the success of the collaboration.

Contents

Chapter 1

Introduction

1.1 Problem Definition

Global Software Development (GSD) is increasingly becoming a common practice in the software industry [19]. Organizations establish global software projects scattered around the globe, involving multiple teams located at different sites. There are many potential benefits that can arise from GSD: lower development costs due to salary savings, decreased development time due to time-zone effectiveness, reduced time to market, and access to the most talented developers [19]. However, since GSD teams are spatially distributed, they have to deal with temporal, geographical, and socio-cultural distances [16], resulting in major difficulties in coordination and communication. Essentially, as Herbsleb suggests [41], "the fundamental problem of GSD is that many of the mechanisms that function to coordinate the work in a co-located setting are absent or disrupted in a distributed project." Research in GSD aims to explore how tools and practices can help dealing with these challenges in order to improve the cooperation across sites.

Traditionally, Software Engineering (SE) deals with processes and tools that help structuring the software development activities. While processes followed in co-located settings vary from traditional waterfall models to agile methodologies, software teams make use of a wide set of collaborative tools for developing software, versioning it, scheduling work, managing requirements and test cases, and sharing knowledge with other team members. However, processes and tools need to be adapted to the teams necessities and adopted by team members. In co-located settings this is supported by planned or impromptu face-to-face meetings, whereas in distributed teams, informal communication cannot take place in person as easily, and the lack of face-to-face communication appears the main obstacle to cooperative practices in GSD settings.

While traditionally, the main media for communication in distributed teams have been email, phone, and video conferencing systems, nowadays communication takes place also in the so-called Social Software (SoSo). Kaplan and Haenlein [52] define SoSo as “a group of Internet-based applications built on the ideological and technological foundations of Web 2.0, that allow the creation and exchange of User Generated Content (UGC).” Essentially, SoSo encompasses a range of software systems that allow users to interact and share information, such as: Instant Messaging (IM), Internet Forums, Blogs, Microblogs, Wikis, Social Network Sites, Social Bookmarking.

With the use of SoSo, it may be possible: to build and keep social relationships between distributed co-workers, to facilitate knowledge sharing inside the organization, to support knowledge management, and to encourage informal communication between distributed team members. Very little empirical research, however, is focused on the use of SoSo in GSD settings, e.g., [2, 37, 99]. Accordingly, this thesis investigates how SoSo complements collaborative tools used in the everyday practices of GSD teams and how it supports remote cooperation. In particular, it explores the role of SoSo for constituting, establishing and maintaining communicative and coordinative practices in GSD.

1.2 Motivation

Traditionally, in Software Engineering (SE), tools and processes help in structuring the software development activities. However, practices in situated action [95] cannot be fully specified by SE methods and processes; often, actual practices differ from plans [83]. Thus, adaptation by teams is necessary: SE methods, tools and processes need to be adopted by the team and adapted to the teams necessities in order to establish shared practices. Whenever team members agree on a set of rules, conventions, and policies — the so-called social protocols [84] — the cooperative activity works smoothly as the team succeeds in achieving and sharing common practices. While face-to-face communication facilitates the negotiation of social protocols in co-located SE, establishing common practices can be seen as one of the main issues in GSD, as we will show in one of the cases studied in this PhD thesis.

In this setting, an important role seems to be played by SoSo to foster, establish, and keep social protocols within a globally distributed software team. SoSo can thus facilitate the adaptation of processes and the adoption of tools, overcoming the lack of face-to-face communication in GSD teams. Previous research [13], [71], [99] has highlighted the role of SoSo in supporting informal communication, e.g., allowing distributed co-workers to build and keep social relationships or facilitating knowledge sharing inside the organization. Moreover, SoSo is a cheap and flexible tool, which is easy to use and familiar to most of developers [7].

SoSo has been successfully adopted in the Open Source (OSS) community for many years, i.e., Wiki, Newsgroups and IRC chats belong to OSS practices e.g., [28], [62], [37]. Therefore, SoSo appears promising to support also software development activities in GSD settings. In particular, the flexibility of SoSo could allow team members to have a flexible channel that can support different communicative necessities, allowing informal communication and complementing other structured traditional SE channels, such as documentation, the ticketing system or the CVS.

1.3 Research Questions and Key Contributions

This thesis aims to contribute primarily to the field of Global Software Development (GSD), combining concepts from Computer-Supported Cooperative Work (CSCW) and Information System (IS) and bringing them to the Software Engineering (SE) community. This research seeks to answer the main research question:

How does Social Software support Global Software Development?

As illustrated in Figure 1.1, the three sub-questions are investigated in sequence. The three sub-research questions addressed are:

(a) How is Social Software used in Global Software Development?

(b) How can we conceptualize the development of communicative and coordinative practices in Global Software Development?

(c) What is the role of Social Software for constituting, establishing and maintaining communicative and coordinative practices in Global Software Development?

In particular, (a) provides insights to formulate an answer to (b) and (b) helps to answer (c). However, (a) supports the answer to (c), and empirical evidences that relate to (c) help to develop the theoretical framework that answers (b), thus also providing new insights in answering (a). The three sub-questions together provide a comprehensive answer to the Main Research Question.

As is evident in Figure 1.1 , the main objective of this thesis is to investigate how Social Software (SoSo) supports GSD. In order to answer this, we perform three ethnographic studies to understand how SoSo is used in real GSD projects (a). During the observation of everyday practices of the GSD teams, the lack of a theoretical framework for analyzing and describing communicative and coordinative practices in GSD became very clear. In particular, the CSCW tradition appears to be useful for investigating the role of SoSo; thus, a theoretical framework built on CSCW concepts is developed in order to answer (b).

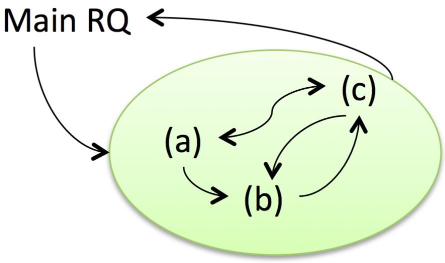


Figure 1.1: The Main Research Question motivating the investigation of three sub-questions.

Thanks to the theoretical framework built to answer (b), it is possible to describe how SoSo supports the constitution, establishment, and maintenance of practices in GSD (c). This also provides insights into the use of SoSo in GSD and thus to the initial sub-question (a). After a comprehensive discussion of the three sub-questions, a detailed answer to the Main Research Question can be provided. In summary, the main contributions of this thesis are as follows:

- *Several empirical contributions mainly related to the central role of SoSo as an informal channel and its key function in complementing collaborative tools used in GSD that provide templates for coordination mechanisms.* SoSo is a flexible tool that supports different kinds of communicative genres. In particular, SoSo enables social talks and metawork, both necessary for establishing and for maintaining effective coordination mechanisms and thus successful collaboration.

- *A theoretical contribution consisting of a theoretical framework for describing and analyzing coordinative and communicative practices in GSD.* The theoretical framework furthers understanding of how SoSo supports cooperation in GSD by combining the theoretical concepts of *coordination mechanisms* and *communicative genres*, both based on the concept of social protocols. This thesis shows that the concepts are complementary in the analysis of the field material and that the theoretical framework is supported by the empirical material derived from the empirical cases.

- *A methodological contribution about the importance of observational studies for understanding cultural, socio-technical, and human aspects of GSD.* In juxtaposition to the actual low adoption of this research method in GSD research, observations can be used for complementing other research methods or triangulating the findings; moreover, they can play a supporting role, as they are used as a preparation for other research methods. This thesis describes the challenges of performing observational studies in geographically distributed settings and suggests how to overcome them.

The remainder of this chapter introduces the research approach adopted and summarizes the three cases and the six papers that constitute this PhD thesis.

1.4 Research Approach

In order to answer the research questions of this thesis, an ethnographically-inspired approach was employed, motivated by a practice-based approach [69] to the research. Practices in situated action [95] often differ from predefined SE methods and processes. Thus, adaptation by teams is necessary: SE methods, tools and processes need to be adopted by the team members and adapted to the teams necessities in order to establish shared practices. A practice-based approach is thus adopted to better understand what is necessary to establish and keep a satisfying cooperation among remote teams in their everyday practices. To understand practices, an ethnographically-inspired approach is employed, assuming the team member perspective: three real world cases have been observed and analyzed cases are summarized in Section 1.5. In particular, data are collected mainly through participant observations, semi-structured interviews, and document analysis. In order to carefully keep track of the investigation, a research diary is kept during each project studied, meetings and interviews are taped and transcribed, and interaction analysis is adapted to examine SoSo logs. By using participant observations, where the researcher participates in the daily routines, it is possible to observe how cooperation, discussions, and conversations take place between team members during the project. Interviews give the possibility to clarify uncertainties and ask about specific issues in a deeper way. Interaction analysis [51] permits analysis in detail of how SoSo is used by team members in their every-day work. Given the multiple ways of collecting data and combining different research methods, it is possible to triangulate the findings and to gain a deep understanding of the work practices. The research approach is further discussed and motivated in Chapter 4.

1.5 Empirical Cases

This thesis is based on three Research Cases:

- **Case 1: DHI.** The final phase of a GSD project in DHI organization [1] was observed for four months; team members were distributed between Denmark, India, and Portland, USA.

[1] http://www.dhi.dk/

- **Case 2: Dispersed agile team.** The project ran for five months and was mainly observed remotely. The team was composed of three team members based in Seattle, two developers in Buenos Aires, and other team members distributed in United Kingdom, Canada, and Netherlands.

- **Case 3: Three student projects.** Each project was composed of two sub-teams located in: Denmark and China, as well as Denmark and Brazil. The project ran for three months, and was observed and analyzed during the whole collaboration.

1.6 Argumentation of the Thesis and Research Papers

Based on the three cases reported in the previous Section, six research papers have been written. Table 1.1 shows an overview of the six research papers, relating them to the research questions they answer, the case they investigate, and the contributions they provide. **Paper 1** is a literature review that provides an initial answer to the Main Research Question, and thus it does not make use of the empirical cases described. **Paper 2** uses the empirical cases as examples to highlight the importance of the observational studies performed: it is a methodological reflection on the research approach adopted in this PhD thesis; thus, it does not aim to directly answer any of the Research Questions proposed. Empirical contributions are reported in **Paper 3**, **Paper 4**, **Paper 5** and **Paper 6**: Case 1 is analyzed and reported in **Paper 3** and re-discussed in **Paper 6**, while Case 2 is described in **Paper 4**. Case 3 is used in **Paper 5** and **Paper 6**. **Paper 6** also provides an answer to the research subquestion (b); it thus includes theoretical contributions. A summary of the six research papers is reported in the following.

1.6.1 Paper 1: Systematic Mapping Study

> R. Giuffrida and Y. Dittrich, "Empirical studies on the use of social software in global software development-a systematic mapping study" Information and Software Technology, 2013.

Paper 1 is a Systematic Mapping Study (SMS) on the use of SoSo in GSD. Since very few empirical papers investigate the specific role of SoSo for the GSD field, the aim of the literature review is not only to map empirical studies on the usage of SoSo in SE projects and in distributed teams but also to highlight the findings beneficial for GSD researchers and practitioners. The paper presents how SoSo is capable of sustaining GSD teams: SoSo is reported as being chiefly used as a support for collaborative work in order to foster awareness, knowledge management, and coordination among team members. Contrary to the

Paper	**Research Question**	**Case**	**Contributions**
Paper 1	Main RQ	—	literature review
Paper 2	—	based on Case 1, 2 and 3	methodological
Paper 3	RQ a and Main RQ	Case 1	empirical
Paper 4	Main RQ	Case 2	empirical
Paper 5	RQ b, c and Main RQ	Case 3	empirical
Paper 6	RQ b, c and Main RQ	Case 1 and 3	empirical, theoretical

Table 1.1: Overview

evident high importance of the social aspects offered by SoSo, socialization is not the most important usage reported. Four emerging themes in GSD are identified: the appropriation and development of usage structures; understanding how an ecology of communication channels and tools are used by teams; the supportive role played by SoSo for socialization; and finally, the surprising low percentage of observational studies.

The SMS motivates this PhD thesis: the identification of the characteristics of SoSo that support SE projects and distributed teams shows that potentially SoSo can be beneficial in GSD settings and that the limited amount of research in this area is due to the novelty of the research field. The SMS highlights interesting aspects that motivate the necessity of further research on the actual use of SoSo in GSD. For example, most of the studies identify a focus on one kind of SoSo at a time, without relating various SoSos to the whole ecology of channels used by the team; this can be justified with the low adoption of observational studies. Without having a broader picture of practices, it is difficult to report how SoSo relates with other channels used in the team. In this PhD thesis, we adopt a practice-based approach to research, employing ethnographically-inspired research methods to gain a deeper understanding of the use of SoSo in GSD. We investigate how SoSo is used in the everyday cooperative practices; thanks to observations, it is possible to relate SoSo to the communicative and coordinative practices of the GSD team. Moreover, we gain a deeper understanding of how SoSo supports social aspects of cooperative work, and thus we can relate its usage to how it helps to overcome some of the challenges of GSD, such as the lack of face-to-face communication.

1.6.2 Paper 2: An Argument for Observational Studies in GSD

Giuffrida, R., and Dittrich, Y. “You can not ask what you do not suspect - an argument for observational studies in global software development research” Submitted to Empirical Software Engineering Journal.

Paper 2 provides an argument for the usage of observational studies in GSD, in juxtaposition with the actual low adoption of this research method in GSD research. This paper presents a mapping of empirical research in the GSD field and reports the research methods adopted to investigate GSD issues. The mapping reveals an extensive usage of interviews and surveys, and a scarce adoption of direct observations for GSD research. An analysis of papers that use observations is performed in order to show the possible advantages of this data collection technique and to motivate the importance of observations in GSD. The paper argues for the necessity of considering observational methods both to more thoroughly explore GSD practices and as a means to triangulate the findings.

The key contributions are: motivating the adoption of observational studies in GSD, defining three kinds of spatial distribution of GSD teams, discussing the challenges of performing observational studies in geographically distributed settings and suggesting how to overcome them.

This paper is mostly based on the experience obtained through the three ethnographically-inspired cases reported in this PhD thesis that are mostly based on observations. The paper strengthens the relevance of the approach adopted in this thesis: through observational studies, it is possible to gain deeper insights about a topic that is not as yet widely investigated and to understand in detail everyday practices of software developers, highlighting socio-cultural aspects involved in the cooperation. Moreover, the researchers reflect in the paper on the experience gained while performing observations of GSD teams and on the challenges encountered due to the geographical distribution of the team members. The principal suggestion is that in GSD most of the cooperation is computer-mediated, and thus the researcher can observe teams remotely, following the approach adopted by team members. Researchers can join team members in the virtual space and visit some of the key sites, as this can lead to gaining a deeper understanding of part of the practices occurring in the physical sites. Finally, the paper clarifies some of the terminology used in GSD settings, as it is evident from the mapping studies performed that there is a lack of shared terminology among researchers about how to refer to the spatial distributions of team members.

1.6.3 Paper 3: Exploring The Role of Instant Messaging

> Y. Dittrich and R. Giuffrida, "Exploring the role of instant messaging in a global software development project" in 6th IEEE International Conference on Global Software Engineering (ICGSE), 2011.

Paper 3 investigates the role of Instant Messaging (IM) in a GSD team distributed across Denmark and India. The paper refers to the DHI case (Case 1) and it explores the integration of formal documentation, bug-tracking systems and email with informal communication on Instant Messaging (IM). Whenever overlap times occur, informal communication can take place at the same time in different sites, and it can effectively complement formal documentation. The analysis provides an indication that IM can play a special role in the socio-technical communication system of the team: IM acts as a real-time glue between different channels. The communication through IM also provides a means to build trust and social relationships with co-workers.

This paper also highlights the dispatcher role of IM chats in the ecology of channels adopted by team members. IM chats support several dimensions of the cooperation - awareness, collaboration, coordination and socialization. Almost every chat analyzed has a social dimension as subtext of the main topic of discussion. The paper suggests the importance of team building, but it does not explain in detail how SoSo supports social relationships between remote team members. This aspect appears interesting and needs to be further investigated in subsequent studies. The paper highlights that there is a lack of analytical tools suitable for understanding the role of IM chats and for explaining the role of IM with respect to other communicative channels that are introduced in the paper and are visible in the field material. Thus, in this PhD thesis, we search for suitable theoretical concepts that can help in understanding practices observed and we then develop a theoretical framework based on such concepts. The findings of this paper are re-discussed in **Paper 6**, indicating how the theoretical framework developed can explain practices described in this paper.

1.6.4 Paper 4: Information Flow Within a Dispersed Agile Team

> H. Sharp, R. Giuffrida, and G. Melnik, “Information flow within a dispersed agile team: A distributed cognition perspective” Agile Processes in Software Engineering and Extreme Programming, 2012.

Paper 4 is based on the dispersed agile team case (Case 2). The paper investigates how the dispersed team collaborates and compares findings with those from co-located agile teams. Using a distributed cognition analysis, the paper describes how information is propagated and transformed within the team. In co-located agile teams, the simple and open flow of information makes use of physical space, it relies on face-to-face communication and on physical artifacts. On the contrary, in the dispersed team, information flow needs to be explicit and the team relies on several digital mediating artifacts containing very detailed information. The study highlights very different patterns of interaction: during the meetings among dispersed team members there is no clear equivalent to the Story Cards and the

Wall. Since team members are dispersed, awareness of each other's activities is not as straightforward as in co-located settings and it is the responsibility of individuals to share information and artifacts with other team members. While in co-located agile teams an important role is played by the social context; in the dispersed agile team under study, a much stronger role is delegated to individuals deciding what to share and with whom.

This paper is written in collaboration with other researchers and it is not based on a case studied for the purpose to answer to the research questions of this PhD thesis. Thus, the paper is not focused on the role of SoSo; however, in the case analyzed, team members make use of IM and Wiki and the paper shows, through the DiCoT analysis, that both kinds of SoSo have a central role in the cooperation. The IM is the chief tool for communication among team members, while the Wiki is the main information hub, where all information is stored. The analysis performed highlights that different kinds of SoSo undertake different roles in the cooperation. However, the distributed cognition approach adopted does not help to understand how to study and describe the role of SoSo in the ecology of channels adopted by the team. Therefore, the paper strengthens the need of finding suitable theoretical concepts that can help to categorize the different usages of SoSo and to better define the relationship between SoSo and the other channels used in the cooperation. In particular, the comparison between co-located teams and distributed ones suggests that, in GSD settings, the social context is disrupted due to the lack of face-to-face communication. Thus, a focus on how team members establish and share social protocols appeares promising in order to understand how cooperative practices are negotiated and maintained in GSD settings, and whether SoSo plays a role in supporting them.

1.6.5 Paper 5: How SoSo supports Cooperative Practices

> Giuffrida, R. and Dittrich, Y. "How Social Software Supports Cooperative Practices in a Globally Distributed Software Project". Cooperative and Human Aspects of Software Engineering (CHASE) Workshop at ICSE, 2014.

Paper 5 analyses one of the teams of the student projects case (Case 3). A detailed analysis of the usage of SoSo during the whole project is conducted using the analytical tools of "communicative genres" and "coordination mechanisms." The paper shows that SoSo is a flexible channel that can enact different kinds of communicative genres and that can support coordination mechanisms, helping GSD team members to deal with spatial, temporal, and socio-cultural distance. Various kinds of SoSos have different roles in the diverse phases of the project: Wiki and Forum are a persistent repository for the knowledge shared; IM is a glue between other channels and is the media where metawork and social talk happen if they cannot take place somewhere else, acting as a dispatcher for other channels; Forum and IM serve as channels for situated articulation that supports established coordination mechanisms. Thanks to the high level of metawork at the beginning of the collaboration and to the social talk, several coordination mechanisms are effectively established; this, alleviating the need of further articulation work. Through SoSo, it is possible to negotiate and establish common social protocols inroder to reach shared practices, allowing that cooperation proceeds smoothly despite the lack of face-to-face communication.

This paper shows that communicative genres and coordination mechanisms are adequate theoretical tools to conceptualize and analyze communicative and coordinative practices in GSD, particularly in relation to the role of SoSo. Even though it requires further discussion, the ecology of channels concept introduced in **Paper 3** is re-used in this paper. In particular, **Paper 5** shows that there is a relationship between coordination mechanisms and communicative genres that deserves further investigation. It appears interesting to explore how communicative genres mediated by SoSo support the establishment, maintaining, and re-negotiation of coordination mechanisms, thus deepening the understanding of the role of social protocols. Therefore, **Paper 6** elaborates on concepts introduced in this paper in order to formulate a more comprehensive theoretical framework, which can be beneficial for future research that aims to analyze and describe not only the role of SoSo but also how cooperative practices are established and maintained in GSD teams.

1.6.6 Paper 6: A Theoretical Framework for Studying the Role of SoSo in GSD

> Giuffrida, R. and Dittrich, Y. "A conceptual framework to study the role of communication through social software for coordination in globally-distributed software teams" Submitted to Information and Software Technology Journal, 2014.

Paper 6 presents a novel conceptual framework for analyzing the role of SoSo in GSD, providing examples from two empirical cases (Case 1 and Case 3) that motivate the development of the framework and explicate it. The framework allows studying the role of communication through SoSo for coordination in GSD. Coordination mechanisms provide a way to collaborate through artifacts, alleviating articulation work. They are, however, supported by explicit communication and by communicative genres that offer a tool to look at computer-mediated communication that occurs between remote teams. The usefulness of the framework is supported by the empirical findings on the role of SoSo, which allows team members to establish, develop, and maintain social protocols during the remote cooperation. In particular, the flexibility of SoSo is highlighted, as it supports different kinds of communicative genres, and it complements coordination mechanisms, assuming a central role in the ecology of channels used by the teams. Moreover, SoSo allows team members to establish, develop, and maintain social protocols during the collaboration. In particular, the importance of social talk through SoSo is highlighted, as not allowing for team building chats may turn out to be expensive.

This paper is motivated by the fact that in literature there is no theoretical tool for researchers to conceptualize and investigate the role of SoSo communication for GSD coordination. In particular, since in GSD most of the activities are computer-mediated, the analysis of coordinative and communicative practices can be performed on the digital artifacts used and produced by the distributed teams. In this paper, cases described in **Paper 3** and **Paper 5** are re-discussed and are used to explicate and to prove the usefulness of the theoretical framework proposed. The framework can be beneficial for future research that aims to analyse and describe how communicative and coordinative practices are established and maintained in GSD teams, showing possible reasons for breakdowns.

1.6.7 Minor Publications

During this PhD project, three minor publications have been submitted and accepted to three Workshops. **Paper 7** and **Paper 8** are published in conference proceedings, while **Paper 9** does not appear in any official proceedings. The workshop papers report partial contributions and work in progress, detailed in succeeding publications. **Paper 10** is a technical report that clarifies terminology used in GSD and is referenced in Chapter 2. References to the Minor Publications are reported in the following:

7. R. Giuffrida, Y. Dittrich, Social Software in Global Software Development. Proceedings of Cooperative and Human Aspects of Software Engineering (CHASE) - Note for CHASE workshop at ICSE, 2010

8. R. Giuffrida, Y. Dittrich, Exploring the Role of Social Software in Global Software Development projects. ICGSE Doctoral Symposium. 2nd Volume of Proceedings of ICGSE, 2011

9. R. Giuffrida, Y. Dittrich, Social Media Ecology in Distributed Workplaces. Workshop on Social Media at Work, ECSCW 2011.

10. P. Tell, R. Giuffrida, H. Shah, Revisiting the Global Software Engineering Terminology. Technical report, 2014.

1.7 Thesis Outline

This thesis is structured in Chapters as follows:

- **Chapter 2** outlines the state-of-the-art in Global Software Development (GSD) and about Social Software (SoSo); furthermore, it summarizes the results from the Systematic Mapping Study on the use of SoSo in GSD (**Paper 1**), highlighting the contributions for research directions in the field.

- **Chapter 3** describes the empirical cases used in this PhD thesis.

- **Chapter 4** presents and discusses the methodology adopted, based on an ethnographically-inspired approach; the chapter includes the discussion carried on in **Paper 2** about the importance of observational studies in GSD.

- **Chapter 5** presents the theoretical underpinnings of this thesis, describing the theoretical concepts used in this PhD thesis: practice, distributed cognition, coordination mechanisms, communicative genres, and social protocols. Moreover, it presents the theoretical framework proposed in **Paper 6**.

- **Chapter 6** discusses the theoretical and empirical contributions of this PhD thesis.

- **Chapter 7** summarizes the main achievements and contributions of the thesis and discusses the current limitations and future work.

Chapter 2

Related Work

2.1 Global Software Development

Global Software Development (GSD) — also called Distributed Software Development (DSD) or Global Software Engineering (GSE) — means splitting the development of the same product or service among globally-distributed sites [57]. There are many potential benefits that can arise from GSD that are promoted to lower development costs due to: 1) salary savings and decreased development time due to time-zone effectiveness and 2) reduced time to market and access the most talented developers [19]. Developing software as a team is a challenging task, but developing software as a global software team is even more challenging due to distances [42]. According to Carmel [15], distance has an impact on the three main forms of cooperation within a team: communication, coordination, and control. Often, in GSD research, dimensions proposed by Carmel are interpreted as "the 3c"; however, other researchers consider other dimensions as "the 3c" [1]. - Fuks et al. [32] refer to communication, coordination, and cooperation, Ellis et al. [29] talk about communication, coordination, and collaboration. Independently from the definition adopted by [29] or [32] or [15], the 3c are widely used in GSD literature to analyze and discuss both tools — e.g., [92] — and practices — e.g., [64], [68]. Since in this PhD thesis, we focus on communicative and coordinative practices, the next subsections report a brief summary of how communication and coordination are currently approached in GSD research.

2.1.1 Communication in GSD

Communication is the exchange between team members of information, whether formal or informal, occurring in planned or impromptu interaction [57]. In distributed teams, the lack of face-to-face communication appears one of the main obstacles for remote cooperation. Software engineers adopt a wide range of communication technologies in addition to, or as

[1] We discussed in detail "the 3c" in **Paper 10**, a technical report that I wrote, together with other colleagues, to clarify some of the terminology used in GSD research.

a replacement for face-to-face communication — e.g., email, mailing lists, telephone, videoconferencing, and not the last, Social Software (SoSo). Despite many theories have been proposed in Computer-Mediated Communication (CMC) literature for media selection [57] — e.g., for Social Presence theory, Media Richness theory, Media Synchronicity theory, etc. — no guidelines based on empirical evidence have yet been established, neither for defining which media is more supportive for GSD communication, nor for which communicative practices are more effective.

Contradictory approaches are proposed and are proven to be beneficial for GSD. For example, Herbsleb [41] reports that communication in GSD is less effective, and he then proposes to work towards compatible processes across sites to reduce the amount of communication necessary to achieve more effective communication. This is in line with other researchers who suggest overcoming GSD challenges by reducing intensive collaboration [15], increasing formal documentation [42], and working on organizational factors such as processes, structure, and goal alignment [16]. However, the success of agile processes in GSD projects points to another direction, as agile methods depend on close collaboration and frequent informal face-to-face communication, rather than lengthy documentation. Some studies suggest that agile practices mitigate GSD challenges: for example, Holmstrom et al. [46] porteeeee specific agile practices to be useful for reducing communication, coordination, and control problems, while Layman [60] proposes that agile methodologies dependent on informal communication can be used on GSD projects, despite geographic, technical, temporal and linguistic hurdles.

These premises motivate the necessity of further studies on tools and practices in GSD, as no standard recommendations are yet available in the field. The use of SoSo in GSD practices might be more wide spread than what is visible in actual research publications. The real challenge lies in answering the question about why and how SoSo can be a useful communication channel for distributed collaboration. Some indications are provided by the success of many open-source projects (OSS) that are coordinated through the wide use of SoSo, such as wikis and instant messaging [57]. Other indications are provided by the Systematic Mapping Study (**Paper 1**) reported in the reminder of this chapter, which motivates the research carried on in this PhD thesis.

2.1.2 Coordination in GSD

Coordination is defined in the Coordination Theory proposed by Malone and Crowston [63] as “managing dependencies between activities.” According to this theory, coordination is achieved by one or more coordination mechanisms: each one addresses one or more dependencies in a situation. Studies in Software Engineering (SE) as well as in Global Software

Development (GSD) often adopt Coordination Theory to investigate software development activities; see e.g., [41, 21, 101]. In the interdisciplinary Coordination Theory, the "mechanism" concept is a general term not related to actual SE practices; thus, in SE as well as in GSD, the concept of coordination mechanism is widely used to indicate a mix of a broad set of practices, methods, processes and tools. For example, in the context of an agile software team [94], tools such as the wiki and the product backlog, activities as the daily standup meeting, as well as roles such as the project manager are all considered coordination mechanisms. Similarly, in the context of a GSD project [16], a set of coordination mechanisms of various natures is analyzed, such as centralized team structure, documentation, periodic commit, communication tools, and periodic meetings.

In CSCW literature, Schmidt and Simone [84] suggest a more rigorous definition of coordination mechanisms based on the use of artifacts for coordination purposes and for alleviating the need of ad-hoc articulation work. Schmidt and Simone agree with Malone and Crowston [63] that cooperative work arrangements have to cope with inter-dependencies of different complexity; however, empirical evidences provided by Schmidt and Simone in the context of cooperative work research show the widespread use of coordinative practices that rely on coordinative artifacts. To the best of the author's knowledge, coordination mechanisms considered with a practice-perspective as suggested by Schmidt and Simone [84] have not been used as yet to analyze GSD practices. However, in GSD, digital artifacts are used for most of the distributed collaboration; thus, an artifact-based view on coordination appears appropriate to analyze and describe GSD practices. Moreover, this definition of coordination mechanisms [84] encompasses concepts such as social protocols and articulation work that appear promising to understand not only coordination in GSD but also the diversity of coordinative and communicative practices, their establishment, and evolution. This discussion will be further detailed in the remainder of this PhD thesis.

2.2 Social Software

Social Software (SoSo) encompasses a range of software systems that allow users to interact and share information. The more specific terms, collaborative software and groupware, are usually applied narrowly to software that enables collaborative work. Distinctions between the terms social and collaborative apply to the applications or uses, not to the tools themselves, although SoSo has been used, albeit rarely, for collaborative work to-date [55]. SoSo includes a wide variety of tools such as: Instant Messaging (IM), Forums, Blogs, Microblogs, Wikis, Social Network Sites, Social Bookmarking. Scholars and practitioners often refer to this set of tools also as Social Media or Web 2.0 tools. A definition of Social Media has been recently developed by Kaplan and Haenlein [52]: “a group of Internet-based applications, built on the ideological and technological foundations of Web 2.0, and that allow the creation and exchange of User Generated Content.” This definition is consistent with the definition of SoSo given by Farkas [30]: tools that (1) allow people to communicate, collaborate, and build community online (2) can be shared, reused, or remixed, and (3) let people learn easily from, and capitalize on the behavior and knowledge of others. The definition by Kaplan and Haenlein [52] implies that content needs to be published either on a publicly accessible website or on a social networking site accessible to a selected group of people: thus, for example, IM is not considered a social media. In this PhD thesis, we refer to SoSo, rather than to Social Media, as in industrial setting we consider the media component — in the sense of broadcasting information in a website publicly accessible — not essential for the internal communication among team members. Thus, we include IM in the SoSo investigated as a channel where informal communication takes place.

2.3 Social Software in Global Software Development

Very few empirical papers investigate the usage of SoSo in GSD. These papers have been collected in **Paper 1**, a Systematic Mapping Study (SMS) on the use of SoSo in GSD: the literature review maps empirical studies on the usage of SoSo in SE projects and in distributed teams, highlighting the findings which could prove to be beneficial for GSD researchers and practitioners. The paper presents how SoSo is capable of sustaining GSD teams: SoSo is reported as being chiefly used to support collaborative work, fostering awareness, knowledge management and coordination among team members. Contrary to the evident high importance of the social aspects offered by SoSo, socialization is not the most important usage reported. From the review, Instant Messaging (IM) is the category of SoSo that is most widely researched in relation to GSD, IM is used in GSD to get quick answers and immediate feedback [59],[98], in order to facilitate multi-tasking and communication with numerous people simultaneously as a secondary channel in meetings [71], and to maintain both a general awareness of the entire team and more detailed knowledge of people with whom they plan to work with [37]. The popularity of IM is confirmed by the work of Black et al. [7], who have run a survey investigating the use of SoSo in GSD. Thanks to their work, it is possible to highlight that IM is not only the most researched SoSo but also the most used among GSD developers. From this mapping study, it emerges that also Wikis are also widely researched in relation to GSD. They are typically used for: organizing and managing documents, storing all project information in just one place and reusing their content, increasing the quality and the completeness of documentation and lowering the barriers for sharing of knowledge [54].

This review has made it possible to provide results from related fields that can give an indication as to the means by which SoSo is capable of supporting virtual teams in GSD. Major difficulties in collaboration, coordination, and communication [16] can be alleviated by the usage of SoSo: IM can replace planned or impromptu face-to-face meetings that are not always entirely feasible in GSD settings; Wiki can be used for knowledge sharing; and Microblogs can generate virtual water-cooler conversations and are used as an informal communication channel [105]. Despite the fact that this review did not reveal a considerable amount of research on Microblogs, this SoSo has been reported as widely used by GSD developers [7]. Other kinds of SoSo, such as Social Networking Sites can be used for socializing [91], project planning, creating team awareness, and for fostering community building and group interactions. They can also help to identify experts in the organization and to get in contact and connect with them in order to gain professional information and knowledge [81], [81], [35]. Social bookmarking can also be beneficial for collaboration in GSD, since it fosters knowledge distribution [22], provides awareness of people's expertise [31] and supports expert finding [22], [31]. Moreover, social bookmarking supports informal processes in software development [99], resource management, information sharing, as well as discovery and social networking [22].

2.4 Systematic Mapping Study Contributions

Paper 1 motivates this PhD thesis: the identification of the characteristics of SoSo that support SE projects and distributed teams shows that potentially SoSo can potwntially be beneficial in GSD settings and that the limited amount of research in this area is due to the novelty of the research field. The SMS andoint out interesting aspects that motivate the necessity of further research on the actual use of SoSo in GSD. Four emerging themes in GSD are identified and addressed, partially or extensively, in this PhD thesis:

1. The socialization role played by SoSo either as a subtext or as an explicit goal.

   Although Pi et al. [75] report positive effects of IM in respect to the satisfaction of both formal and informal communication in the work context, socialization or usage of SoSo for non-work related topics is seldom reported. This might depend on the specific research design and research question, as some papers [18], [73] indicate that SoSo can be used explicitly to facilitate and maintain relationships. In most of the research, though, the social dimension seems to show as a "subtext" to work-related interaction, confirming findings of **Paper 3**. To cite an example, Handel and Herbsleb report about a project in which the content of IM chats is primarily focused on work task, with a smattering of non-work topics and humor [39]. Other works report similar findings [80, 3]. Due to its perception as channel for informal communication, SoSo seems to invite more social subtexts than other media consequently supporting social relations and team spirit. A better understanding of the role of social relationships and their development as part of any successful collaboration may improve and augment the future design of tools and may encourage the adoption of SoSo in GSD practices so as not to hinder informal communication and social subtext.

2. The appropriation and development of usage structures.

   The often cited advantage of SoSo as providing an unstructured communication channel also appears to provide a challenge. Throughout the reviewed articles, the need to develop structures, rules, good practices or agreements on how to use SoSo in work contexts is mentioned (see, e.g., [97], [96],[74], [47], [40], [36], [56], [79]). SoSo appears to afford conscious appropriation by teams and organizations, perhaps because of its lack of structure. Social protocols [84] that establish the use of SoSo need to be negotiated and maintained within the project or organization. This appropriation in turn results in tailored support for the specific project rather than requiring the team to adjust to predefined structures. The co-construction of social protocols guiding the usage of tools and communication channels needs to be better understood. This might in turn help to understand the appreciation of SoSo tools by software engineering practitioners.

3. Understanding how an ecology of communication channels and tools are used by teams.

   Although few of the studies address more than one kind of SoSo, many remark upon the necessity of examining more than one communication channel and the interaction among these various channels when they are used by a team [97] [98],- [74], [100],[47], [106], [80], [14], [79]. Turner et al. [100] propose using the concept of “communication ecology” and consider SoSo part of a set of tools that complements other available communication channels. The authors also highlight the role of different SoSo in this ecology. Bradner et al. use the term “interaction ecology” when analyzing the adoption of the IM tool in IBM [11]. When analyzing the use of SoSo, many studies refer to other tools, but they do not discuss in detail: 1) how different communication channels complement each other and 2) the complementary role of different channels. By providing unstructured channels, SoSo appears to possess the potential to both coordinate the use of other channels as well as to provide a media to negotiate communication channel usage. It is in this field that researchers can focus to understand how SoSo and other tools can be used effectively.

4. The need for more observational studies.

   The quantitative analysis of research methods applied in the SMS indicates that only 11 of the studies included in the review used observational methods as part of their research. The three emerging themes discussed above pertain to varied issues that are difficult to grasp through mere analysis of activity traces such as logs, interviews and surveys, or experiments. They entail the requirement tof relating the use of SoSo to the context in which it is used: other communication channels, the affordances of the development approach and project model used, among others. Studies such as this thesis will provide a cogent understanding of the interplay between technical affordances and the social protocols developed, when adopting the tools in a specific context. Research conducted along these lines is important to support future design and appropriation of collaboration support. Moreover, it is possible to gain a deeper understanding of how SoSo supports social aspects of cooperative work by relating its usage to how it helps to overcome some of the challenges of GSD, such as the lack of face-to-face communication. It would appear advisable and worthwhile to take a step back and concentrate additional efforts on the understanding of the distributed practices before focusing on new tools and process support.

Research addressing the four themes identified will not only provide more information on the usage of SoSo but will also aid in making enhanced support available for GSD and distributed development. This PhD thesis aims to provide contributions in these research directions.

Chapter 3

Empirical Cases

This thesis is based on three research cases. Please note that I personally collected and analyzed data as the main researcher in Case 1 and Case 3. However, I participated only in the analysis of Case 2, in which the main researcher was Prof. Helen Sharp [1], this resulted from a collaboration in the fall of 2011 during a visiting period at the Open University [2].

3.1 Case 1: DHI

DHI is an independent, international, consulting and research organization. The company develops and uses high-end hydraulic simulation software. We observed part of the World Bank Project (WB-Project), which has a considerable amount of software development. This is a global distributed project: five members are settled in Copenhagen, Denmark; seven members in Delhi, India and one Project Area Manager in Portland, USA. The Danish team is composed of one project manager and four Project Area Managers (PAM), the Indian team consists of five developers, one team leader, and one tester. We observed the team while working on the development of a Decision Support System (DSS) for water management in the Nile Basin. The observations took place during the final part of the development process of the first release of the system mainly during the testing phase of the same release. The project was successful, and the team has subsequently developed a second release of the software. The ethnographic empirical research took place both in Copenhagen and in Delhi in the fall of 2010. I collected field material and, in order to have a better understanding of the practices, observed the team for four months from Denmark and also spent two weeks in India.

[1] Helen Sharp is a Professor of Software Engineering in the Computing Department of The Open University (UK) where she leads the Empirical Studies of Software Development research group. http://mcs.open.ac.uk/hcs2/

[2] http://www.open.ac.uk/

3.2 Case 2: Dispersed agile team

The project under study was to develop enterprise software components for use by software developers in their own organizations when building their cloud-based solutions. The deliverables are composed of binaries, tests, and developer guidance. The team was a successful partially-dispersed agile team. The development, including spiking iterations, ran for five months from July 2011 to November 2011. The project team consisted of one core team of nine members, an additional offshore testing team, and a network of advisers (which was globally dispersed). The core team was composed of three team members based in Seattle, two developers in Buenos Aires, and other team members distributed in United Kingdom, Canada, and Netherlands. Most team members had worked together on few previous projects, and hence knew each other. At critical times within the product development, all team members who could attend would visit Seattle for one or two sprints. The main researcher, Prof. Helen Sharp, conducted an observational study of the agile team, observing all team members remotely. Additionally, USA and UK sites were visited in order to enhance the understanding of the team.

3.3 Case 3: Three student projects

The three student projects under study are part of a GSD student cluster in collaboration between IT University of Copenhagen (ITU), Peking University (PKU) and Universidade Federal de Pernambuco (UFPE). Each team is formed by two remote sub-teams. One team (Team A) has participants located in Denmark and in Brazil, while the other two teams (Team B and Team C) have participants located in Denmark and in China. The collaboration took place from February 2011 to May 2011. In each team, a supervisor provided the description of the product to be developed and he evaluated the work performed by the students. The system design, the requirement specifications, the development of the product and the organization of the collaboration were students' responsibilities. Therefore, the teams were "self-organizing" their work, sharing roles, responsibilities and decision taking. I attended most of the meetings from the Danish side and visited PKU teams for a week, collected pictures, took notes during observations of meetings and interviewed participants.

Chapter 4

Research Method

This Chapter describes the research method adopted to investigate the three cases that constitute the empirical material for this PhD thesis. The research is carried on through empirical Software Engineering (SE) techniques. Given the topic we are focusing on, qualitative methods are the most suitable for our research [26]. In particular, an ethnographically-inspired approach has been adopted in the three cases delineated in Chapter 3. This chapter illustrates empirical qualitative software engineering methods, motivates the usage of this approach and summarizes how the research has been conducted. The last part of the chapter highlights the challenges encountered and summarizes methodological contributions of this PhD thesis, based mainly on the findings reported in **Paper 2**.

4.1 Empirical Qualitative Software Engineering (SE)

Empirical research methods are well-established as part of the SE research method toolkit. After developing and establishing quantitative empirical research in the late 80s and beginning of the 90s [5], the discussion of qualitative research methods started in 2000 with the workshop "Beg, Borrow, or Steel: using multidisciplinary approaches in empirical software engineering research" [88]. Landmarks of this discussion are special issues on Qualitative Research in Software Engineering [24] and the Workshop series on "Cooperative and Human Aspects of Software Engineering" [1], since 2008 at the International Conference on Software Engineering (ICSE) [2]. Qualitative empirical methods have been promoted to be able to understand software development as social practices.

According to Seaman [85], qualitative research methods are needed to study nontechnical aspects of software engineering. In many other disciplines, qualitative research methods have been developed and are commonly used to handle the complexity of issues involving human behavior [85]. Several qualitative methods for data collection and analysis are available

[1] http://www.chaseresearch.org

[2] http://www.icse-conferences.org/

and can be incorporated into empirical studies of software engineering, also in combination with quantitative methods. For a list of possible data collection techniques, see the work by Easterbrook et al. [26].

4.1.1 Ethnographically-inspired Approach

Ethnography is a form of research studying a community of people to understand how the members of that community make sense of their social interactions [82]. In Software Engineering, ethnography can help to understand how technical communities build a set of practices that enables them to perform technical work collaboratively [26]. An ethnographic approach avoids imposing any pre-existing theories: it focuses on how the members of the community themselves make sense of their social and cultural setting [26]. An ethnographic study is conducted through field observations, with a researcher collecting field notes in a research diary; the researcher explicitly considers his own pre-conceptions and how they influence his understanding of the studied community. Underlying ethnographic research is the idea that "members of a community construct their social and cultural practices on the fly, and their perceptions of those structures also define them" [82]. Thus, ethnographic researchers do not seek to prove hypotheses and theories, but rather create local theories to improve the understanding of the practices observed [26].

The main research question of this thesis — *How does Social Software support Global Software Development?* — aims to investigate cooperative practices of GSD teams, that also involve social aspects. Ethnographic studies in SE are valuable for discovering what goes on in particular technical communities, and for revealing subtle but important aspects of work practices [26]. For this reason, an ethnographically-inspired approach [3] has been employed to study three different cases with the purpose of understanding how team members make use of SoSo for accomplishing their work in distributed settings. In particular, the set of data collection techniques described in Section 4.2 has been employed for all cases. Two cases consist in industrial real-world projects, while one case is a student project, which was meant to simulate a real world GSD project and has thus been treated like other cases. The same approach has been adopted in order to thoroughly investigate practices in all cases.

4.2 Data Collection Techniques

We collect data through different empirical techniques: non-participant observation, semi-structured interviews and document analysis. These techniques will be briefly described in

[3] The approach is referred to as ethnographically-inspired because this researcher is not a professional ethnographer. Although the study has been conducted in a rigorous manner, sociologists may not consider it to be pure ethnographic due to the limited time dedicated to the observations and to the limited level of involvement of the researcher as a non-participant observer in the project.

the following paragraphs.

4.2.1 Observations

The term observations can spam over a wide range of methods: situated and online observations; short term and long term observations; and participant and non-participant observations; in short, any research method including observations of real world software development. In-situ observations are typically documented by fieldnotes, but they can also be captured on audio or video — see the work conducted by Boden et al. [9] as an example in GSD. In-situ observations permit getting data about "real life" in the real world [82]. Observational studies might be necessary to gain deeper insights or to triangulate findings when used in combinations with other research methods, as described in detail in **Paper 2**. However, observations are neither easy nor trouble-free: they require a heavy investment of time and effort [82]. Moreover, observations can be considered inappropriate for software development as it can be considered an activity that takes place inside a person's head [85]; however, software developers reveal their thought processes when communicating with other developers [77], and thus as a cooperative work [24] Software Engineering can be likewise observed.

4.2.2 Interviews

Interviews ask respondents about facts or about their individual understanding of a situation. Interviews are conducted with a variety of objectives: they are used to collect historical data from the memories of interviewees, to collect opinions or impressions, or to help identify the terminology used in a particular setting [85]. Interviews can be structured or unstructured; many studies employ semi-structured interviews [85]. Interview data are often captured on audio or video, and recorded in field notes. Interviews can be subject to respondent bias; they depend on the interviewer formulating questions about the area of concern, and the interviewee might not be able to account for what actually happened.

4.2.3 Document Analysis

A document is any substance that gives information about the investigated phenomenon and exists independently of the researcher's actions. Documents consist of real world data, physical or digital, that provide traces of collaboration, e.g. in form of manuals, emails, chat logs, wiki pages, code structures, as well as in-situ participatory observations. Documents are normally produced for specific purposes other than those of the research, but they can be used by the researcher for studying the past or for gaining deep insights about the project [20]. Robson [82] points out advantages and disadvantages of document analysis:

an advantage is that documents are unobtrusive and can be used without imposing on participants; they can be checked and re-checked for reliability. A major problem is that documents may not have been written for the same purposes as the research and therefore conclusions will not usually be possible from document analysis alone.

4.2.4 Multi-method Approach

In this PhD thesis, data were collected through different empirical techniques: non-participant observations, semi-structured interviews, and log analysis[4]. In order to carefully keep track of the investigation, a research diary of observations was kept during each project observed. The researcher taped and transcribed meetings and interviews, and then qualitatively and quantitatively analyzed the content of informal computer-mediated communication through SoSo. By using non-participant observations, where this researcher participated in the daily routines of the teams (as a researcher), the type of collaboration that took place between team members in the different projects[5] was observed. Semi-structured interviews gave the possibility of clarifying uncertainties and deepening the understanding of specific issues. Log analysis was performed through interaction analysis[51] and using the theoretical framework developed in this thesis — see Section 4.3 for a description of the Data Analysis techniques. Workshops were organized in Case 1 to summarize the outcomes and to support researchers and practitioners in reflecting together on the findings obtained. By using multiple ways of collecting data and combining different kinds of methods, it was possible to triangulate the findings [82].

4.3 Data Analysis

The data analysis of Case 1 and Case 3 was mostly performed on the content of Social Software (SoSo) logs — Instant Messaging (IM), Forum and Wiki through interaction analysis [51]. Interaction analysis, from a sociolinguistic perspective, is concerned with the structure of social interaction manifested in the conversation. It emphasizes features of social context and is used for analyzing the exchange of messages among team members; bottom-up dimensions are applied as codes in the analysis of SoSo logs. In our study, the coding schema

[4] Particular reference is made to Case 1 and Case 3. Case 2 was performed through an ethnographically-inspired approach for which the same data collection techniques were used; however, since the study was conducted by another researcher and I participated only in the data analysis phase, minor differences occurred in the approach adopted. See Chapter 3 for the description of the three cases.

[5] As a non-participant observer, I sat together with participants, observing their daily routines, but I was not part of the team as a technical member. Observations lasted for some weeks, and thus only for a limited time. A purely ethnographic approach would usually require months and intense participation of the researcher in the daily work.

was developed while coding the messages; it was then discussed among the researchers involved. Results of the analysis were triangulated with the examination of other documents, such as mail, technical documents, and logs of collaborative tools, compared with collected field notes subsequently verified through semi-structured interviews with team members.

The analysis of Case 2 was mainly performed on documentation, Wiki, IM logs, and on field notes through the distributed cognition framework. Afterwards, the analysis was used to formulate structured interviews that provided a deeper understanding of the project and helped to triangulate the findings obtained. The distributed cognition analysis was meant to analyze the flow of information in the team that is not focused on understanding the role of SoSo. Thus, distributed cognition did not appear sufficient to understand the role of SoSo in the ecology of channels used by the team. For this reason, a novel theoretical framework was developed and adopted to analyze the role of SoSo in Case 1 and Case 3. The novel conceptual framework was used in combination with interaction analysis to classify SoSo logs in Case 3 and re-analyze the IM chats of Case 1, thereby identifying several communicative genres [6].

4.4 Challenges

There are two main challenges in this PhD project [7]: 1) performing ethnographic observations in globally distributed projects that require to observe practitioners in different workplaces, and 2) gaining access to SoSo logs such as IM logs, sometimes available only in the personal computer of team members. The first challenge appears particularly interesting from a methodological point of view: traveling across countries to have an understanding of what is happening in each location appears necessary, and reflections on this have been reported in **Paper 2**, which motivates the usage of observational studies in GSD and suggests how to perform them when dealing with geographical distribution — see Section 4.5 for a synthesis of the discussions carried on in the paper. The second challenge is related with other well-known challenges of qualitative studies [77], such as being allowed to collect data from documents, gaining access to companies or dealing with trust acquisition. Fortunately, in all projects, for both the student and industrial cases, the access to field material was not an issue, and participants were generally willing to provide the researchers with the material needed. In some circumstances, not all IM logs were obtained; however, the

[6] See Chapter 5 for a detailed description of the distributed cognition approach and of the novel theoretical framework used for the analysis that includes the communicative genre analytic tool.

[7] This Chapter does not discuss the challenges of qualitative studies in general or the challenges of ethnographic research, which can be viewed as one of the challenges of this PhD thesis. Some discussions on this topics are reported in **Paper 2**. For further references, see the works by [85], [26], [77]

richness of qualitative data collected allowed the describing of practices despite the lack of some interactions.

4.5 Methodological Contributions

In this PhD thesis, a methodological contribution is offered by the novel approach adopted that combines interaction analysis [51] with the identification of communicative genres in SoSo logs. With interaction analysis, it is possible to identify fine-grained genres that describe carefully the communication occurring in SoSo, providing an understanding about the social context of the teams. Moreover, further contributions are provided by the adoption of the ethnographically-inspired approach in GSD settings. Since ethnography involves a considerable amount of observations, and observational studies are not widely adopted in GSD settings, a methodological reflection about the approach used was performed in **Paper 2**, providing several arguments for the usage of observational studies in GSD. In particular, the paper highlights that:

1. there are few observational studies in GSD research, which is mostly carried out through surveys and interviews; however, observations complement other research methods and are crucial in GSD research to catch human and social aspects of GSD teams;

2. it is possible to deal with the geographical distribution of team members; thus, the paper provides recommendations on how to perform observational studies dealing with geographical distance.

These contributions are detailed in **Paper 2**, which maps empirical research methods adopted to investigate GSD issues and provides an argument for the usage of observational studies in GSD. This is in contrast with the actual low adoption of this data collection technique in GSD research. Moreover, it argues for the need of more observational studies in GSD, rather than relying on interviews and surveys as the primary — and most widely adopted — method of investigation, as often "researchers cannot ask what they do not suspect." Observations can be used for complementing other data collection techniques or for triangulating the findings (equal role). They can also have a supporting role; in other words, used as a preparation for other research methods, e.g. observing the project in a preliminary phase to gain initial insights that will be further investigated through other research methods, such as interviews or surveys. Observations are particularly decisive in GSD research, since they permit the investigation of cultural, social, and human aspects of

GSD projects. This is shown in the paper through detailed examples, taken mostly from the three Cases employed in this PhD thesis.

Paper 2 provides another methodological contribution, suggesting that the primary challenge in performing direct observations in GSD is the geographical distribution of participants: the researcher cannot be physically present in different sites at the same time. Taking inspiration from the work of sociologists for ethnographic studies in global settings [65], [66], [58], the paper describes how researchers can deal with geographical distance while observing GSD projects. In particular, multi-site ethnography [65] and virtual ethnography [43] are indicated as inspirational approaches. Researchers need to take into consideration the different kinds of distribution of the teams — dispersed, distributed, or partially-dispersed — and in relation to this, the choice of the appropriate sites to observe. It can be necessary to visit all sites, to choose to observe only the most strategically-relevant ones, or to include the online environment as the sole — or additional — site to observe. In particular, an ideal approach would include several observations, complementing virtual observations with physical ones. Moreover, it is possible to observe remote sites together with digital traces of collaboration: the observations of what is happening in the physical sites can be complemented with what is digitally happening among team members - virtual meetings, documents exchange, communication through SoSo, emails, etc[8]. In this way, it is possible to get a comprehensive understanding of what is going on in the distributed project, considering the different perspectives involved.

[8] Please note that document analysis can be considered as part of the observations of the virtual site, as most of the distributed collaboration takes place through online traces and documents

Chapter 5

Development of a Conceptual Framework

5.1 A Practice-based Approach

The previous chapter illustrated the research method adopted in the present PhD thesis: ethnographically-inspired methods have been chosen as a suitable approach for answering the research questions of this thesis. As in other studies that make use of ethnographic research in Software Engineering (SE) — e.g., [87], [78] — we are interested in revealing the actual work practices of people in GSD projects and in "studying this socio-technical phenomenon in its natural state" [87]. This is in contrast to the approach of traditional SE as a discipline, which is "mainly concerned with the formal principles, the technical basis and the methodological support for software development, rather than the reflection of software practice as a human activity that goes beyond the engineering framework" [23]. Often, practices in situated action [95] differ from pre-defined SE methods and processes. Detailed descriptions of practices provide the basis for understanding the social factors [23] that influence effective practices [53], as well as for designing better tools [4]. Practice-based approach studies are becoming more widespread in SE research — e.g., [53], [70] — as well as in GSD research — e.g., [87], [9]. This is in contrast with other approaches for designing better tools, such as the cognitive approach, widely used in human-computer interaction to study the cognitive processes of programmers — i.e., the mental processes involved in programming [89].

A practice-based approach [69] is adopted in this PhD thesis in order to understand and describe actual practices in three GSD cases. Several examples adopting this approach are available in GSD: Sigfridsson and Sheehan [87] describe an Open Source community called PyPy; Boden et al. [10] examine the coordination practices in distributed software development of small enterprises; and Avram [4] investigates a project with the purpose of better understanding collaborative work and knowledge management processes in distributed software development settings. These research papers are examples of adopting ethnography as a research approach to study work practices; however, they suffer in proposing an effective way to analyze such practices. For example, Boden et al. [10] indicate the concept of *articulation work*, as defined by Strauss [93], as their analytic lens for data collection and analysis. Although the articulation work concept is relevant also in our context, it is not sufficient alone to describe cooperative practices in GSD. During this PhD project, we recognized — as shown in detail in **Paper 6** — the lack of a proper analytic framework for understanding and describing communicative and coordinative practices in GSD settings.

In CSCW tradition, the facilitating role of artifacts in collaborations is well recognized. For example, Star and Griesemer [90] introduce the concept of boundary objects as a theoretical construct to understand the role of artifacts. In her study on boundary negotiating artifacts [61], Lee adopts ethnographic methods to understand how a team of designers use physical artifacts and social practices to collaborate: artifacts play a role in the active negotiation of shared understanding among communities of practice. Through the negotiation process in which professionals interact with artifacts, it is possible to achieve local and temporary alignment of social practices [103]. This is in line with Schmidt and Simone's research [84] that introduces the concept of coordination mechanism, consisting in "a social protocol imprinted upon a distinct artifact [...] that alleviates the need for ad-hoc articulation work" [84]. Coordination mechanisms are adopted in the theoretical framework proposed in this thesis, rather than other concepts, such as boundary objects. This is because the coordination mechanism concept combines together both artifacts — that can be considered as boundary objects, which are "incredibly important to collaborative work" [61] — and social protocols — which underpin the negotiation and alignment of practices [86], [103]. These concepts will be further explored in the remainder of this chapter and in Chapter 6, that discusses the findings of this PhD thesis.

5.2 Theoretical Underpinnings

This section describes the different analytical concepts used in the empirical papers that constitute this PhD thesis. The analysis of the field material was conducted adopting different conceptual approaches in order to describe practices of GSD teams in relation with their usage of SoSo. The different attempts led the researchers not only to develop a novel theoretical framework based on some of the concepts illustrated in this section, but also to describe and analyze communicative and coordinative practices in GSD.

5.2.1 Overview

In **Paper 3**, we categorize Instant Messaging chat logs occurring among team members defining dimensions visible in the field material and adopting a bottom-up perspective; no conceptual tools are used for analyzing the SoSo logs. Although social protocols and coordination mechanisms are visible in the field material, a comprehensive view of the role of SoSo within the ecology of channels used by the team is missing. In **Paper 4**, a distributed cognition [48] approach is used to analyze the information flow in a GSD agile team. Despite the visibility of the supportive role of SoSo in the information flow of the team, the approach adopted does not help in conceptualizing the role of SoSo. Thanks to the two approaches adopted, there is apparent evidence of the lack of an effective instrument to: analyze SoSo logs and to relate SoSo with other tools; conceptualize and describe the role of SoSo in GSD teams; and in particular, to relate communicative and coordinative practices. For this reason, in **Paper 5**, we relate the coordination mechanism concept with the communicative genres as an analytic tool, finding interesting matches among the two approaches and recognizing a promising support for defining the role of SoSo within the ecology of channels used by GSD teams. Therefore, **Paper 6** proposes and explicates a novel conceptual framework, based on the concepts of coordination mechanisms, communicative genres and social protocols to code SoSo logs and to understand the role of SoSo in the ecology of channels used by different GSD teams. All analytical concepts used in the papers are briefly summarized and discussed in the following. The theoretical framework is presented in Section 5.3.

5.2.2 Distributed Cognition

Distributed cognition theory [48] examines the cognitive processes that are dispersed among individuals and between individuals and artifacts in the external environment with and through that they interact [45]. Within CSCW and HCI, it has been used to investigate collaborative working — e.g., [33], [38]. This kind of analysis views a cognitive system as being distributed across individuals, artifacts, internal (i.e., cognitive) representations, and external representations in the environment. It focuses particularly on how information is propagated and transformed within the system to achieve collaboration.Despite being related to cognitive theories, distributed cognition is compatible with the practice-based approach adopted in this PhD thesis, as it investigates what occurs in the external environment, rather than what is in the mind of the participants, and tries to get knowledge about the dispersed cognitive system by observing tangible artifacts and the physical environment.

A technique called DiCoT has been used in **Paper 4** to analyze the team's information flows. DiCoT [8] provides a structured approach for reasoning about a situation from a distributed cognition point of view. It draws on ideas and representations from contextual design [6], together with a series of principles that are central to distributed cognition. There are three main themes in DiCoT:

1. The physical theme focuses on the physical environment within which the cognitive system operates at whatever level of granularity is relevant, from the building or office layout to the positioning of items on a desk or noticeboard.

2. The artifact theme focuses on the detail of artifacts that are created and used to perform the activity under study.

3. The information flow theme focuses on what and how information flows through the cognitive system, the media which facilitate that flow, and how the information is transformed in the process.

Furniss and Blandford [8] identify 22 principles from distributed cognition which can be loosely categorized according to these three themes. In **Paper 4**, we investigate each theme using these principles, providing a description of the practices observed in relation with the information flow within the team. Thanks to the DiCoT analysis performed, it has been possible to highlight the role of Wiki as an "information hub," a central focus where information flows meet and decisions are made, and the important role of IM, a media used for synchronous communication. The DiCoT analysis helps to understand how each SoSo used is related to other tools adopted in the team. However, as in GSD most of the collaboration is computer-mediated, the analysis also reveals a weak impact of the physical theme, as it does not provide interesting insights into understanding remote collaboration; it rather highlights the fundamental role played by the digital artifacts. Moreover, DiCoT describes how the information flows in the cognitive system, not describing how the system is established and how practices are negotiated and developed. These aspects are instead supported by the coordination mechanism analytical tool introduced in **Paper 3** and adopted in **Paper 5**; the coordination mechanism is based on the concepts of social protocols and artifacts. Finally, DiCoT does not give the opportunity to deepen the analysis of the communication occurring in SoSo, that is, it does not provide a tool for analyzing and describing communicative practices in detail. Therefore, communicative genre analysis has been introduced in **Paper 5** to classify communication through SoSo.

5.2.3 Communicative Genres

Genre Theory states that genres create order to simplify the mass of available information [17]. A genre of organizational communication is characterized by a socially recognized communicative purpose and a common form [104]. The communicative purpose of a genre "is not rooted in a single individual's motive for communicating, but in a purpose that is constructed, recognized and reinforced within a community" [67]. The form of a genre refers to the "readily observable features of the communication, including structural features, communication medium and language" [72]. Structural features can be text formatting, such as lists and headings, as well as devices for structuring interactions at meetings, such as agenda and chairperson; the communication medium can be pen and paper, face-to-face, a telephone or a mail; the language can be the level of formality or the specialized vocabulary used [72]. People produce, reproduce and change genres through a continuous process of negotiation and re-adaptation [104]. Genres may be considered at different levels of abstraction and they can be combined in genre repertoire, a set of genres routinely enacted by a particular community. A community's genre repertoire "indicates its established communicative practices and it can serve as an analytic tool for investigating the establishment of a community's communicative practices" [72]. Im et al. [49], for example, use genre and genre repertoire for analyzing email communication of a geographically dispersed software team; in their work, they analyze and describe in detail the genres identified in the electronic communication among team members.

Inspired by the work of Yates and Orlikowski [72] that widely investigate the notion of communicative genres as a way for structuring communicative practices in organizations, we consider that the communicative genre is an appropriate analytical tool to understand and describe the communication occurring through SoSo in GSD. Thus, in **Paper 5** we adopt this analytic concept to describe SoSo communication in a GSD team — showing its suitability in providing interesting insights on the role of SoSo. We thus decided to use it as a basis for the theoretical framework presented in **Paper 6**. However, we considered the concept of purpose with a different meaning than the one presented by Orlikowski and Yates [72]. In their work, they identify purposes such as: response, question, proposal, for your information (FYI) and meta-comment. These purposes are not topic-related and are not sufficient for the detailed analysis of concrete interaction through SoSo. In our framework, the concept of purpose of a communicative genre is considered with a contextual meaning, rather than the one presented by Orlikowski and Yates in their studies [72]. Since, as mentioned above, communication genres can be identified at different levels, we consider content related purposes — such as awareness and team building — in order to understand the roles of communicative genres within collaborative software development practices. The response/question dimension used by Orlikowski and Yates [72] is maintained as a generic categorization that, when appropriate, specifies sub-genres with more specific purpose within the content related purposes identified.

5.2.4 Coordination Mechanisms

In CSCW literature, Schmidt and Simone [84] suggest a rigorous definition of coordination mechanisms:

> "A *coordination mechanism* consists of a *coordinative protocol* imprinted upon a distinct *artifact* which [...] *stipulates* and *mediates* the articulation of cooperative work so as to reduce the complexity of *articulation work* [...]" [84] (emphasis in original)

The coordinative protocol consists of a set of rules e.g. taken-for-granted ways of proceedings, established conventions, official policies, standard operating procedures, while the coordinative artifact is a "stable data structure expressed in a standardized graphical format" [84]. Schmidt and Simone report, for example, about a bug report form[1], a two page form (the artifact) with several fields filled by different actors, which follows a set of agreed procedures and conventions (the protocol) and which stipulates the responsibilities to the different roles, the possible classifications of bugs, reports of bugs corrected, etc. The artifact is "the distinct and persistent symbolic construct in which the protocol is imprinted and objectified" [84]. The concepts of articulation work and social protocols are described in the following subsections.

Several examples of coordination mechanisms are presented in **Paper 5** to complement the communicative genre analysis performed. A promising mutual supportive role among the two concepts is visible in the field material, thus the two concepts have been combined in a theoretical framework presented in **Paper 6**. This is in line with the necessity arisen in **Paper 3** and **Paper 4**, to relate SoSo communication with other channels used in the teams, in order to understand the relationship between SoSo and other tools and artifacts.

Articulation Work

Schmidt and Simone discuss the concept of articulation work that is defined as:

> "[...] a recursive phenomenon in that the management of an established arrangement of articulating a cooperative effort may itself be conducted as a cooperative effort which, may also need to be articulated" [84].

In cooperative work settings characterized by complex task interdependences, "coordination mechanisms reduce the complexity of articulation work and alleviate the need for ad hoc deliberation and negotiation" [84]. We distinguish two levels of articulation work, as

[1] The article was published before issue trackers were wide spread in industry.

suggested by Gerson [34] and Strauss [93]: metawork and situated articulation. Metawork is used to describe the development of a protocol, while situated articulation denotes the articulation of the state of the current task in order to coordinate this task; situated articulation involves "adapting a social protocol to a situated use" [76]. Metawork and situated articulation can be understood as communicative genres that are used for discussing, modifying and establishing social protocols and coordination mechanisms.

Articulation Work is reported in the analysis in **Paper 5** as the purpose of some communicative genres identified in SoSo. However, the relationship between articulation work genre — both metawork and situated articulation — and coordination mechanisms is detailed in **Paper 6**, through examples taken from the empirical material.

Social Protocols

The definition of coordination mechanisms explicitly mentions the role of the social protocol: a set of rules, conventions, and policies shared by people involved in the cooperative activity. In the communicative genres definition, protocols are not defined; however, it is reported that "social norms are visible in recurrent communicative situation" [104], thus, social norms are underpinning also the concept of communicative genres. More in general, the concept of social protocol (or social norm or social rule) is related with the concept of practice and in particular with the social nature of it, as stated by Wittgenstein [102]:

> [...] *obeying a rule* is a practice. And to think one is obeying a rule is not to obey a rule. Hence it is not possible to obey a rule *privately*: otherwise thinking one was obeying a rule would be the same thing as obeying it. (PI nr 202 p. 81) [102] (emphasis in original)

Thus, the social nature of the norm comprises the necessity to be shared by people involved. Wittgenstein also suggests that [102]:

> [...] we lay down rules, a technique, for a game, and that then when we follow the rules, things do not turn out as we had assumed (PI nr 125 p. 50) [102]

That is to say, social protocols need not only to be initially decided upon, but also adapted and adopted by people over time, thus socially shared, modified, negotiated, and appropriated.

Paper 6 shows that social protocols come about and are maintained in different ways. Social protocols can be both explicitly and implicitly defined by team members. A social protocol is explicitly defined through metawork when team members discuss whether and how to use a specific artifact. For example, a team can decide to share an agenda (the artifact of the coordination mechanism) prior to a meeting and that each sub-team is alternatively responsible for that. A social protocol can be implicitly adopted — e.g., based on previous work experience in the project or from professional knowledge. The social protocol on how to use the agenda during the meeting might be implicitly driven by previous experience of team members and does not require further discussion on how to use the artifact. A social protocol can also be defined but not adopted, when team members agree on it but later do not use it. **Paper 6** provides several examples on how social protocols are formed, developed, negotiated, and established by team members of different GSD teams.

5.3 Theoretical Contribution: a Conceptual Framework

Paper 6 present a framework for describing and analyzing coordinative and communicative practices in GSD projects, relating the notions of communicative genres and coordination mechanisms described in the previous paragraphs. Both notions are mutually supportive and are based on the concept of social protocols that influence practices and evolve as the project progresses. Changes in the actual work can cause changes of the social protocols. The framework proposed is reported in Figure 5.1 and it aims to describe how SE process models and methods are adapted in the actual practices by GSD teams.

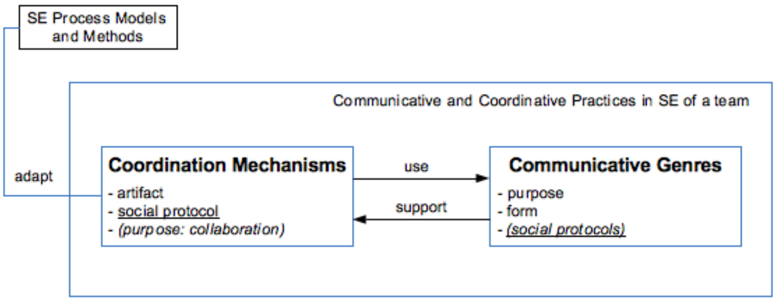


Figure 5.1: Conceptual Framework

Since in GSD most of the activities are computer-mediated, the analysis of coordinative and communicative practices can be performed on the digital artifacts used and produced by the distributed teams. To the best of the author's knowledge, coordination mechanisms as defined by Schmidt and Simone [84] have not been used as yet to analyze GSD practices. However, in GSD, digital artifacts are used for most of the distributed collaboration; thus, an artifact-based view of cooperation appears appropriate to analyze and describe GSD practices. Coordination mechanisms provide a way to collaborate through artifacts, alleviating articulation work. However, they are supported by explicit communication, and thus by communicative genres that offer a tool to investigate computer-mediated communication that occur between remote teams.

In summary, the theoretical framework provides a tool that allows researchers and practitioners:

1. to investigate the role of SoSo in the remote cooperation;
2. to relate the use of SoSo with the ecology of channels (repertoire) used in the GSD team;
3. to analyze and describe how social protocols are defined, negotiated and established among team members through computer-mediated communication;
4. to analyze and describe communicative and coordinative practices in GSD that differ from pre-specified processes, models and methods.

The theoretical findings will be further discussed in Chapter 6. For a detailed description of the Conceptual Framework and relative examples, see **Paper 6**.

Chapter 6

Findings

Chapter 1 introduced the Research Questions that this PhD thesis aims to answer, describing how they relate to the six Research Papers that constitute the thesis and with the contributions provided. For ease of reference, this thesis aims to answer to the following Main Research Question:

> How does Social Software support Global Software Development?

Three sub-questions are further addressed:

(a) How is Social Software used in Global Software Development?

(b) How can we conceptualize the development of communicative and coordinative practices in Global Software Development?

(c) What is the role of Social Software for constituting, establishing and maintaining communicative and coordinative practices in Global Software Development?

In order to answer to the research questions, three empirical cases have been studied — cases are described in Chapter 3 — and, as indicated earlier, the following six Research Papers have been written:

- **Paper 1** - R. Giuffrida and Y. Dittrich, “Empirical studies on the use of social software in global software development — a systematic mapping study” Information and Software Technology, 2013.
- **Paper 2** - Giuffrida, R., and Dittrich, Y. “You can not ask what you do not suspect — an argument for observational studies in global software development research” Submitted to Empirical Software Engineering Journal.
- **Paper 3** - Y. Dittrich and R. Giuffrida, “Exploring the role of instant messaging in a global software development project” in 6th IEEE International Conference on Global Software Engineering (ICGSE), 2011.

	Paper 1	**Paper 2**	**Paper 3**	**Paper 4**	**Paper 5**	**Paper 6**
Main Research Question	X		X	X	X	X
Research Question (a)			X	indirect	indirect	indirect
Research Question (b)					X	X
Research Question (c)			indirect	indirect	X	X

Table 6.1: Relationship between Research Questions and Research Papers. In the table, an X indicates that an answer is directly provided in the original papers. However, indirect answers can be derived and are reported in the discussions carried on in the remainder of the present chapter.

- **Paper 4** - H. Sharp, R. Giuffrida, and G. Melnik,"Information flow within a dispersed agile team: A distributed cognition perspective" Agile Processes in Software Engineering and Extreme Programming, 2012.
- **Paper 5** - Giuffrida, R. and Dittrich, Y. "How social software supports cooperative practices in a globally distributed software project" Cooperative and Human Aspects of Software Engineering (CHASE) Workshop at ICSE, 2014.
- **Paper 6** - Giuffrida, R. and Dittrich, Y. "A conceptual framework to study the role of communication through social software for coordination in globally-distributed software teams" Submitted to Information and Software Technology Journal, 2014.

Table 6.1 shows to which Research Question the different Research Papers provide an answer. All papers answer the Main Research Question, except for **Paper 2**, which consists of a reflection about the observational studies performed in this thesis; the methodological contributions are discussed in Chapter 4. **Paper 1** has been partially discussed in Chapter 2 and it consists of a literature review; however, it is included in the discussions presented in this chapter, as it provides a preliminary answer to the Main Research Question. **Paper 3**, **Paper 4**, **Paper 5** and **Paper 6** directly or indirectly answer the different sub-questions. A direct answer to the specific sub-questions is addressed in the original papers; however, considering the whole thesis, further discussions can be provided; thus, indirect answers are as well discussed in the following, complementing the findings reported in the original papers.

6.1 Answers to the Research Questions

This section provides a preliminary answer to the Main Research Question — mainly considering the findings of **Paper 1** — and detailed answers to the subsequent sub-questions that give additional insights to the Main Research Question.

How does Social Software support Global Software Development?

An initial answer to the Main Research Question is directly provided by the Systematic Mapping Study (SMS) presented in **Paper 1**, which consists of the literature work; further contributions are presented in Chapter 2. In brief, the SMS shows that major difficulties in GSD related to collaboration, coordination, and communication can be alleviated thanks to SoSo, which is reported as being chiefly used as a support for collaborative work, fostering awareness, knowledge management, and coordination among team members. The SMS shows that few research studies are focused on the role of SoSo in GSD; however, it collects empirical papers about the usage of SoSo in distributed teams and in SE projects, suggesting how SoSo can be used in GSD. For example, IM can replace planned or impromptu face-to-face meetings [71], [59] that are not feasible in GSD settings; Wiki can be used for knowledge sharing [54]; microblogs can generate virtual water-cooler conversations and can be used as an informal communication channel [105]. Further examples are reported in **Paper 1**. Four emerging themes become visible thanks to the SMS:

1. The socialization role played by SoSo either as a sub-text or as an explicit goal: SoSo seems to invite more social subtext than other media and, consequently, supports social relations and team spirit; however, the usage of SoSo for non-work related topics is seldom reported. A better understanding of the role of social relationships facilitated by the usage of SoSo and their development as part of successful collaboration is further explored in this PhD thesis, in particular in **Paper 3**, **Paper 5** and **Paper 6** and will be detailed in the following while answering to research sub-question (a);

2. The appropriation and development of usage structures: since SoSo is an unstructured communication channel, there is a need to develop structures, rules, good practices or agreements on how to use SoSo in work contexts; thus co-construction of social protocols guiding the usage of such communication channel and other tools need to be better understood; we addressed this aspect in **Paper 5** and **Paper 6**, as described in the answer to sub-question (c);

3. Understanding how an ecology of communication channels and tools are used by teams: the complementary role of SoSo and how it relates with other channels need to be better understood; in **Paper 6** we proposed to adopt a theoretical framework based on the concept of repertoire to explain the relationship between SoSo and other tools used in the team, as explained while answering to sub-question (c);

4. The need of more observational studies: thanks to observational studies, it is possible not only to gain a deeper understanding of how SoSo supports social aspects of cooperative work, but also to analyze how it helps to overcome some of the challenges of GSD. Providing a detailed description of the actual use of SoSo in real-world GSD projects, in particular in relation with other channels, is the purpose of the whole thesis, and it is discussed while answering to the three following sub-questions. A detailed argumentation for the importance of observational studies is reported in **Paper 2**.

In summary, **Paper 1** provides an initial answer to the Main Research Question and raises four emerging themes that have been addressed in this PhD thesis. In the following, an answer to each sub-question is provided in order to complement the preliminary answer to the Main Research Question offered by the Systematic Mapping Study presented in **Paper 1**.

(a) How is Social Software used in Global Software Development?

Paper 3 has been written to provide an answer to sub-question (a); however, **Paper 4**, **Paper 5** and **Paper 6**, that have been written subsequently to answer to other research questions, also offer answers to this sub-question, adding insights informed by the progresses of the research obtained during the advancement of the PhD project.

In particular, **Paper 3** shows that, whenever overlapping hours occur, nearly synchronous communication through Instant Messaging (IM) can take place across different sites; thus, thanks to IM, real-time collaboration is possible. As a flexible channel, IM allows work discussions, i.e., chats about code, requirements, tests, and other artifacts. In IM chats, people clarify what is formally stated in other channels, complementing the information shared in a mail or in an issue tracker system. Thus, IM effectively complements other communication channels and traditional collaboration tools, acting as a glue between different channels and as a dispatcher for the communication occurring in the team. IM is a fundamental part of the ecology of channels used in the team — i.e. the socio-technical system where different channels and tools are used in a complementary way. Four dimensions are visible in the chat logs: coordination, collaboration, awareness, and socialization. Although only one dimension is usually predominant, the social dimension is often present as a sub-text. The social sub-text supports the collaboration and helps to keep good relationships between remote team members.

Paper 4 describes how the information flows within a GSD agile team; thus, the focus of the paper is not on how SoSo is used in a GSD team. However, thanks to the distributed cognition analysis performed, the usage of Wiki and IM has been described, therefore the paper partially answers sub-question (a). According to the distributed cognition analysis, the Wiki is used as an "information hub" — i.e., a central focus where information flows meet and decisions are made — together with other traditional tools, such as the issue tracker system and the shared repository. All team members communicate freely with all others, and synchronous communication and collaboration is possible, mainly through the use of IM and phone calls. Communication is mainly informal, and team members are easily accessible for impromptu conversations over IM. Some team members report that they collaborate on a daily basis with some remote colleagues, even for several hours per day; documentation (e.g., shared digital artefacts, wiki pages, and recordings of the meetings) is limited to what is considered necessary for getting the work done, as in every agile project. As SoSo is a flexible tool, it appropriately accomplishes this "ad-hoc" function, supporting impromptu informal communication.

The usage of Wiki and of IM observed in the agile team described in **Paper 4** is in line with some research works [71], [59], [54], as reported in **Paper 1**. Additionally, the distributed cognition approach describes how the information flows in the ecology of channels used by the team, offering a complementary perspective to the one adopted in **Paper 3** and filling the gap identified in **Paper 1**, in which other channels and tools are not considered when investigating the usage of SoSo. However, the distributed cognition approach does not appear sufficient to describe the role of SoSo within the ecology of channel chosen by the team — i.e., showing the relation with other communicative channels and collaborative tools — or to provide a detailed description of the usage of SoSo by team members — i.e., describing which kinds of communication is carried on and whether socialization occurs through SoSo. Therefore, in **Paper 5** the analytical tools of communicative genres and coordination mechanisms have been adopted to describe in more detail the role of SoSo as part of the ecology of channels, as well as to investigate the usage of SoSo as a flexible tool that supports different communicative genres.

Paper 5 shows that IM, Wiki and Forum are flexible tools that support different communicative genres, including work discussions, knowledge sharing, team building, and articulation work. It is interesting to note the support of SoSo in the establishing phase of the project, when decisions need to be taken, social relationships need to be established, and social protocols need to be negotiated. After the initial phase, Forum and Wiki serve as a persistent repository for the knowledge shared, while the communication through Forum and IM allows situated articulation, decision making, and collaboration record. IM is the media where things happen if they cannot take place somewhere else, acting as a dispatcher for other channels and as the main channel for social talk. SoSo enables social talk and metawork, both fundamental for establishing and maintaining successful collaboration. On the one hand, social talks support the actual work in distributed environments by enhancing the development of social relationships among team members. On the other hand, metawork is fundamental for the software development activities, allowing negotiating and establishing of social protocols in the collaboration. Providing to distributed teams the access to flexible tools such as SoSo, and encouraging social talks appear to have an impact on the success of the collaboration.

The importance of social talks, arisen in **Paper 3**, is confirmed in **Paper 5**, which shows additional dependencies with other communicative and coordinative practices occurring in the team, in particular with the negotiation and establishment of common social protocols. Therefore, to understand in depth the usage of SoSo and to relate it to the ecology of channels used in different teams, the necessity of a more comprehensive conceptual framework emerges; this has been extensively discussed in **Paper 6** and is described in the following sub-question.

(b) How can we conceptualize the development of communicative and coordinative practices in Global Software Development?

In answering the previous research question, it appears necessary to find a way to conceptualize the development of practices in GSD, particularly in relation to the concept of social protocols [84]. As highlighted in **Paper 1**, the appropriation and development of usage structures are important when studying SoSo in GSD. Because of its lack of structure, SoSo requires co-construction of social protocols guiding the usage of such communication channel and of other tools. **Paper 3** provides initial indications about the importance of social protocols in an established GSD team, though it does not investigate this aspect in detail, **Paper 3** and **Paper 4** reveal the lack of an appropriate conceptual tool that allows to describe GSD practices and the role of SoSo associated to those practices. Since in GSD most of the activities are computer-mediated, the analysis of GSD practices can be performed on the digital artifacts used and produced by the distributed teams during the remote cooperation. Thus, an artifact-based concept such as coordination mechanism [84], which includes the notion of social protocols, appeared appropriate to analyze and describe GSD coordinative practices. However, being SoSo a communication channel, the analytical tool of communicative genre [72] is introduced to investigate communicative practices that support and complement coordination mechanisms. Both analytic concepts have been adopted in **Paper 5** to study and describe the development of communicative and coordinative practices in a novel GSD student team. The two concepts appear compatible and mutually supportive, and thus they have been integrated in a comprehensive theoretical framework.

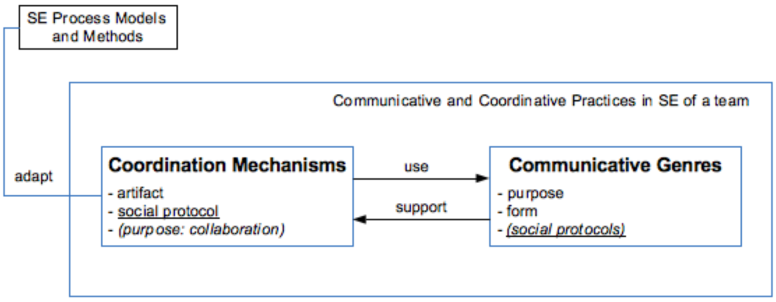


Figure 6.1: Conceptual Framework described in detail in **Paper 6**.

Paper 6 presents the novel conceptual framework, as reported in Figure 6.1, integrating the analytical tools of coordination mechanisms and communicative genres. On the one hand, coordination mechanisms provide a way to collaborate through artifacts, alleviating articulation work. On the other hand, coordination mechanisms are supported by explicit communication, thus by communicative genres that allow to investigate computer-mediated communication occurring between remote teams. Both notions are mutually supportive and are founded on the concept of social protocols that influence practices and evolve as the project progresses. The conceptual framework offers to researchers a tool for looking at communicative and coordinative practices in GSD projects. In particular, the framework aims to describe how SE process models and methods are adapted in practice by GSD teams, as practices in situated action [95] cannot be fully specified by SE methods and processes, and thus adaptation by the teams is necessary.

The theoretical framework has been developed while analyzing the field material of Case 3; however, it also serves to extend the understanding of the role of SoSo in Case 1 and Case 2, that was analyzed prior to the development of the framework. Thanks to the framework, it is possible to deepen the understanding of how SoSo is used in GSD — complementing the answer to sub-question (a) — and to explain the role of SoSo in relation to other channels, i.e., the role of SoSo within the ecology of channels used by the team and the supportive role of SoSo for establishing and maintaining social protocols. These aspects are described while answering the following research sub-question.

(c) What is the role of Social Software for constituting, establishing and maintaining communicative and coordinative practices in Global Software Development?

Paper 1 highlights the lack of research works focusing on understanding the relationship between SoSo and the ecology of tools used by GSD teams and the need of investigating co-construction of social protocols guiding the usage of SoSo and of other tools. These topics are addressed in the following.

SoSo is part of an ecology of channels that has to be explored as a whole, not exclusively focusing on the specific functionality of each kind of SoSo. It appears necessary to first analyze and illustrate the ecology of channels used by the teams and, subsequently, the role of SoSo can be described within the ecology. The ecology of channels concept presented in **Paper 3** is explained by the framework presented in **Paper 6** through the concept of repertoire of communicative genres and of coordination mechanisms. That is to say, any coordination mechanism or communicative genre is part of a repertoire (or ecology) that can be — and has to be — studied as a unitary system in order to understand the relationships between different channels and tools. **Paper 6** highlights the central role of SoSo as an informal channel and its key function in supporting traditional SE tools that function as coordination mechanisms. Moreover, it shows that every team develops its own arrangement of communication channels and tools (ecology of channels or repertoire); however, common aspects can be observed across teams, e.g., that the same communicative genres are visible in different teams and the supportive role of SoSo for enacting coordination mechanisms. Communicative genres and coordination mechanisms are mutually supportive: SoSo enables social talks and metawork, both necessary for establishing and for maintaining effective coordination mechanisms, and thus result in successful collaboration. In particular, SoSo enables a channel for metawork and articulation work. Metawork serves to initiate a coordination mechanism; once the coordination mechanism is defined and adopted, the social protocol can be re-discussed and redefined by team members through further metawork. Thus, the social protocol of the coordination mechanism evolves over time, thanks to metawork. Situated articulation takes place as part of employing the coordination mechanism, supporting and enhancing it.

The theoretical framework proposed is founded on social protocols, which need to be discussed, developed, and established in any SE project. On the one hand, when direct communication can take place, team members explicitly negotiate and agree on social protocols. On the other hand, social protocols can develop implicitly as the project progresses and changes while practices evolve. In particular, SoSo allows developing, negotiating, and maintaining social protocols among team members, thus overcoming the lack of face-to-face communication. Its usage differs whether the team is established or not. Thanks to the conceptual framework presented in **Paper 6**, it is possible to describe how social protocols are formed, negotiated, established and maintained by team members in GSD projects and how they are supported by SoSo. Social protocols need to be shared and appropriated by team members in order to achieve common understanding — e.g., on how to use tools or on how to cooperate with remote colleagues. The importance of negotiating and agreeing on common social protocols is particularly decisive in GSD settings, in which direct communication cannot take place so easily — as it is often mediated by artifacts — and in which socio-cultural distance can affect the collaboration. Thus, agreeing on social protocols becomes even more challenging and crucial. The analysis through the theoretical framework of different GSD teams reported in **Paper 6** shows the dynamics of how social protocols are negotiated and established within novel GSD teams. It provides an indication of why breakdowns occur, and it describes how social protocols are maintained in an established team. In all cases, the supportive role played by SoSo for constituting, establishing, and maintaining social protocols is highlighted, and it is related to the maintaining and developing of informal relationships among team members.

6.2 In Summary

The aim of this PhD thesis is to answer the following Main Research Question:

> How does Social Software support Global Software Development?

In order to answer this question, a description of different usages of SoSo has been reported, based on a Systematic Mapping Study and on three empirical cases, highlighting the flexibility of SoSo that supports different communicative purposes. The field material, however, suggests the necessity of a comprehensive conceptual framework in order to deeply understand the usage of SoSo and to relate it to the ecology of channels used in different teams. The conceptual framework is based on the concepts of communicative genres and of coordination mechanisms and it offers a tool to analyze and describe computer-mediated GSD practices, especially focusing on the role of SoSo. Thanks to the theoretical framework, the central role of SoSo is highlighted: on the one hand, it supports metawork to enact and negotiate coordination mechanisms; on the other hand, it supports different kinds of communicative genres, e.g., work discussions, knowledge sharing, articulation work, and team building. SoSo complements traditional SE tools that function as coordination mechanisms, supporting informal communication. Using the theoretical framework developed, it is possible to describe when and how the metawork through SoSo is successful and whether or not it succeeds in establishing common coordination mechanisms. Moreover, the theoretical framework furthers the understanding of how and why metawork and coordination mechanisms fail. The success of the cooperation strongly relates to how coordination mechanisms and social protocols are established among team members, helping to bridge language barriers, cultural differences, and time and space distances. The development of satisfying communication and coordination practices seems to be dependent on the social cohesion of the team developed through encouraging communication and social talks.

Chapter 7

Conclusions

7.1 Contributions

This thesis explores the role of Social Software (SoSo) for supporting Global Software Development (GSD) teams. There are three main contributions of the thesis: empirical, theoretical, and methodological.

From an *empirical* perspective, this thesis presents the central role of SoSo as an informal channel and its key function in complementing collaborative SE tools. SE tools provide templates for coordination mechanisms, while SoSo is a flexible tool that supports different kinds of communicative genres, including work discussions, knowledge sharing, team building, and articulation work. SoSo allows team members to establish, develop, and maintain social protocols during the collaboration. Informal communication shared through SoSo as a side channel complements professional coordination mechanisms. Providing distributed teams the access to flexible tools such as SoSo and encouraging team building communication impact on the successful establishment of coordination mechanisms. Not allowing for cheap talk might turn out to be expensive.

From a *theoretical* perspective, this thesis proposes a theoretical framework for analyzing communicative and coordinative practices in GSD. The framework combines concepts derived from Computer-Supported Cooperative Work (CSCW) and Information System (IS) traditions in order to understand and describe GSD cooperative practices, and it offers researchers a tool to analyze computer-mediated cooperative practices in distributed settings. The framework is based on, and combines, the concepts of communicative genres and coordination mechanisms. Communicative genres with the purpose of articulation work can initiate, redefine, enhance, and support coordination mechanisms, which alleviate the necessity of further and more complex articulation work. Other kinds of communicative genres, such as work discussions, knowledge sharing, and team building, support the collaboration, thus sustaining the establishment of coordination mechanisms. The success of the collaboration strongly relates to how coordination mechanisms and social protocols are established among team members, helping to bridge some GSD issues, such as language and technological barriers or professional differences.

From a *methodological* perspective, this thesis is strongly based on direct observations [82] of GSD team members. Observational studies allow obtaining a more comprehensive understanding of work practices when implicit behavior is to be captured, as in GSD, where cultural, socio-technical, and human aspects are essential part of everyday practices. Investigating such aspects is challenging, as they are often not accessible solely through surveys and interviews. For this reason, in this thesis we make extensive usage of direct observations as a data collection technique within an ethnographically-inspired approach. The thesis contributes in providing an argument for a wider adoption of observational studies in GSD, in juxtaposition with the actual low usage of this kind of research method in GSD research. Moreover, it discusses the challenges of performing observational studies in geographically distributed settings and suggests how to overcome them — see **Paper 2**.

7.2 Implications for Research and for Practice

This thesis showed how situated action [95] can be analyzed to study communicative and coordinative practices in GSD teams. Practices in situated action are not fully specified by SE methods and processes. Thus, adaptation by the teams is necessary. The concept of social protocols as part of both communicative genres and coordination mechanisms helps to further explore these processes. A theoretical framework based on the notion of social protocols is useful: to understand breakdowns, to investigate the establishment of practices in novel GSD teams, and to analyze the re-negotiations of practices in established teams. The theoretical framework can be beneficial for future research that aims to analyze and describe not only the role of SoSo, but also how communicative and coordinative practices are established and maintained in GSD teams.

An encouraging contribution, both for researchers and for practitioners, is that the success of the collaboration strongly relates to how coordination mechanisms and social protocols are established among team members, helping to bridge language barriers, cultural differences, time and space distances. The analysis indicates the importance of communication supported by SoSo as a side channel that complements professional coordination mechanisms. It does not appear necessary to design new tools, as it is very important to understand how existing tools can be adopted and what the potentials of their usage are. It seems advisable for practitioners to ensure not only that metawork and social talk are supported by the communication channels used by team members but also that they take place, as geographical distribution does not preclude the possibility of having these kinds of conversations. Our framework allows the description of the heterogeneity of the repertoire in different teams; however, it does not aim to be exhaustive in having identified all needs for achieving a successful remote collaboration. Metawork and social talk appear to be part of these needs. Future research could investigate in detail what the elements are that have to be part of a repertoire to make it effective. A list of needs could be provided to practitioners to verify whether all necessary elements — such as, but not exclusively, metawork and socialization — take place somewhere in the repertoire.

Research in GSD has established the importance of informal communication for cooperation in distributed settings — e.g., [27], [15], [50]. Before the development of the theoretical framework, we affirmed that SoSo supports informal communication in **Paper 3** and in **Paper 4**; thus, SoSo can be used to complement formal SE tools. With the repertoire perspective proposed in the theoretical framework, the distinction between formal and informal does not seem appropriate any longer, as all communicative genres have a recognizable form and purpose. Thus, a tool itself is not formal or informal, while both formal and informal communication can take place, e.g., in IM chat as well as in mail. What can make the distinction could be the accountability towards the management and towards the outside. For example, an issue tracker system could be consider formal, as its usage is explicitly defined and it provides information that can be used from outside, e.g., by customers. Social protocols on how to use an IM chat, however, are not explicitly defined, and IM is used only for internal communication in the team. These reflections can support researchers that aim to further investigate the role of “informal” communication in GSD.

7.3 Limitations and Future Work

The main limitation of the present work is that this thesis describes cooperative practices without suggesting actions or providing strong recommendations. Despite implications for research and practices being visible in the empirical material, guidelines have not been suggested and dedicated evaluation has not been performed; thus, our findings are limited to the description of observed practice. Through action-research methods, such as cooperative method development [25], it would be possible, after the initial observational phase, to suggest how to improve practices and to evaluate ad-hoc improvements. We are convinced that research was not mature enough to undertake this line of research prior to this PhD thesis. However, thanks to the contributions of the Systematic Mapping Study we performed — see **Paper 1** — and to the description of practices in the three GSD cases, a basis for further investigation of the role of SoSo in GSD teams has been provided. Further research can embrace findings of the present PhD thesis and can explore other kinds of GSD teams — e.g., larger teams — or it can focus on one specific case and attempt to modify practices, introducing or improving the use of SoSo, in order to evaluate how it helps to overcome GSD barriers.

Another limitation of this thesis is that, being based on three qualitative cases, it is challenging to claim that findings can be extended from our cases to a larger population — e.g., in projects involving larger teams or more sub-teams spread in several time zones. However, the deep analysis offered by our qualitative approach shows that empirical findings can be generalized in the theoretical contributions we propose. The theoretical framework works in three different cases with distinct characteristics, and thus researchers can use the framework also in other contexts to analyze cooperative practices through the coordination mechanism and communicative genre concepts. Undoubtedly, the framework can be extended while studying further cases: additional kinds of coordination mechanisms and communicative genres can be identified, and thus new categories could be found and integrated in the framework. Moreover, the framework is focused on communicative and coordinative practices; however, in SE, further practices can be taken into consideration and analyzed. For example, peculiarities of the different code practices could be included in an extended version of the framework, using specific analytical concepts. Similarly, design practices, documentation practices as well as testing practices could be included, finding suitable conceptualizations.

Another aspect that has not been investigated in detail and would deserve further investigation is how cultural issues affect GSD practices. Culture could have a role on the use of SoSo and on how social protocols are established. However, from our field material, national culture [44], traditionally used to study and to explain cultural differences, does not seem to deeply affect the collaboration. Especially in the student projects, where comparisons between the three teams can be performed, the professional culture and the personal attitudes of people appear to be more influential than is the national culture: some teams achieved a negotiated working culture [12] that permitted team members belonging to different countries to effectively work together, while other teams did not achieve this common shared understanding. In this context, an important role seems to be played by some team members (cultural liaison) in helping the two teams to overcome distance and to foster a sense of cohesion among the members, allowing metawork and team building chats to lower possible cultural barriers, such as the language barriers, that are the only ones observable in the cases studied. However, these themes have only been touched in a marginal way in **Paper 5** and would deserve further attention in future works.

Finally, another theme has been briefly discussed and would deserve further investigation: the role of trust in the collaboration among team members. The importance of trust has been highlighted in literature, as it fosters effective communication and team collaboration, e.g. [1]. Despite the field material showing the importance of social aspects, we have not investigated in detail the relationship between team building and effective work; we have also not explained a possible cause-effect relationship. From our field material, it is visible that team building helps to create effective relationships among team members, facilitating the negotiation and establishment of common practices. However, we have not proven the relationship as we have not performed any psychologically controlled experiment to show that when people exchange cheap talk, they are more efficient. However, we have shown that there is a relationship between team building and good work practices, facilitated and supported by the usage of Social Software. This aspect would also deserve further investigation in future work.

Bibliography

[1] Ban Al-Ani and David Redmiles. Trust in distributed teams: Support through continuous coordination. *Software, IEEE*, 26(6):35–40, 2009. — Cited on page 53.

[2] Ofer Arazy, Ian Gellatly, Soobaek Jang, and Raymond Patterson. Wiki deployment in corporate settings. *Technology and Society Magazine, IEEE*, 28(2):57–64, 2009. — Cited on page 2.

[3] Daniel Avrahami and Scott E Hudson. Communication characteristics of instant messaging: effects and predictions of interpersonal relationships. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 505–514. ACM, 2006. — Cited on page 19.

[4] Gabriela Avram. Of deadlocks and peopleware-collaborative work practices in global software development. In *Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on*, pages 91–102. IEEE, 2007. — Cited on page 30.

[5] Victor R Basili. The role of experimentation in software engineering: past, current, and future. In *Proceedings of the 18th international conference on Software engineering*, pages 442–449. IEEE Computer Society, 1996. — Cited on page 23.

[6] Hugh Beyer and Karen Holtzblatt. *Contextual design: defining customer-centered systems.* Access Online via Elsevier, 1997. — Cited on page 33.

[7] Sue Black and Joanne Jacobs. Using web 2.0 to improve software quality. In *Proceedings of the 1st Workshop on Web 2.0 for Software Engineering*, pages 6–11. ACM, 2010. — Cited on pages 2, 17, and 18.

[8] Ann Blandford and Dominic Furniss. Dicot: a methodology for applying distributed cognition to the design of teamworking systems. In *Interactive Systems. Design, Specification, and Verification*, pages 26–38. Springer, 2006. — Cited on pages 32 and 33.

[9] Alexander Boden, Gabriela Avram, Liam Bannon, and Volker Wulf. Knowledge management in distributed software development teams-does culture matter? In *Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on*, pages 18–27. IEEE, 2009. — Cited on pages 25 and 30.

[10] Alexander Boden, Bernhard Nett, and Volker Wulf. Coordination practices in distributed software development of small enterprises. In *Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on*, pages 235–246. IEEE, 2007. — Cited on pages 30 and 31.

[11] Erin Bradner, Wendy A Kellogg, and Thomas Erickson. The adoption and use of 'babble': A field study of chat in the workplace. In *ECSCW'99*, pages 139–158. Springer, 1999. — Cited on page 19.

[12] Mary Yoko Brannen and Jane E Salk. Partnering across borders: Negotiating organizational culture in a german-japanese joint venture. *Human relations*, 53(4):451–487, 2000. — Cited on page 52.

[13] Michael J Brzozowski. Watercooler: exploring an organization through enterprise social media. In *Proceedings of the ACM 2009 international conference on Supporting group work*, pages 219–228. ACM, 2009. — Cited on page 2.

[14] Ann Frances Cameron and Jane Webster. Unintended consequences of emerging communication technologies: Instant messaging in the workplace. *Computers in Human Behavior*, 21(1):85–103, 2005. — Cited on page 19.

[15] Erran Carmel and Ritu Agarwal. Tactical approaches for alleviating distance in global software development. *Software, IEEE*, 2001. — Cited on pages 14, 15, and 51.

[16] Marcelo Cataldo, Matthew Bass, James D Herbsleb, and Len Bass. On coordination mechanisms in global software development. In *Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on*, pages 71–80. IEEE, 2007. — Cited on pages 1, 15, 16, and 18.

[17] Daniel Chandler. An introduction to genre theory. *The Media and Communications Studies Site*, 1997. — Cited on page 33.

[18] Hee-Kyung Cho, Matthias Trier, and Eunhee Kim. The use of instant messaging in working relationship development: A case study. *Journal of Computer-Mediated Communication*, 10(4):00–00, 2005. — Cited on page 18.

[19] Eoin Ó Conchúir, Pär J Ågerfalk, Helena H Olsson, and Brian Fitzgerald. Global software development: where are the benefits? *Communications of the ACM*, 52(8):127–131, 2009. — Cited on pages 1 and 14.

[20] Piergiorgio Corbetta. *Social research: Theory, methods and techniques.* SAGE Publications Limited, 2003. — Cited on page 25.

[21] Kevin Crowston, Hala Annabi, James Howison, and Chengetai Masango. Effective work practices for software engineering: free/libre open source software development. In *Proceedings of the 2004 ACM workshop on Interdisciplinary software engineering research*, pages 18–26. ACM, 2004. — Cited on page 16.

[22] Laurie E Damianos, Donna Cuomo, John Griffith, David M Hirst, and James Smallwood. Exploring the adoption, utility, and social influences of social bookmarking in a corporate environment. In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, pages 86–86. IEEE, 2007. — Cited on page 18.

[23] Yvonne Dittrich, Christiane Floyd, and Ralf Klischewski. *Social thinking, software practice.* The MIT Press, 2002. — Cited on page 30.

[24] Yvonne Dittrich, Michael John, Janice Singer, and Bjørnar Tessem. Editorial: for the special issue on qualitative software engineering research. *Information and software technology*, 49(6):531–539, 2007. — Cited on pages 23 and 25.

[25] Yvonne Dittrich, Kari Rönkkö, Jeanette Eriksson, Christina Hansson, and Olle Lindeberg. Cooperative method development. *Empirical Software Engineering*, 13(3):231–260, 2008. — Cited on page 51.

[26] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. Selecting empirical methods for software engineering research. In *Guide to advanced empirical software engineering*, pages 285–311. Springer, 2008. — Cited on pages 23, 24, and 27.

[27] Kate Ehrlich and Marcelo Cataldo. All-for-one and one-for-all?: a multi-level analysis of communication patterns and individual performance in geographically distributed software development. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 945–954. ACM, 2012. — Cited on page 51.

[28] Margaret S Elliott and Walt Scacchi. Free software developers as an occupational community: resolving conflicts and fostering collaboration. In *Proceedings of the 2003*

international ACM SIGGROUP conference on Supporting group work, pages 21–30. ACM, 2003. — Cited on page 3.

[29] Clarence A Ellis, Simon J Gibbs, and Gail Rein. Groupware: some issues and experiences. *Communications of the ACM*, 1991. — Cited on page 14.

[30] Meredith Farkas. *Social software in libraries: building collaboration, communication and community online.* Information Today, Inc., 2007. — Cited on page 17.

[31] Stephen Farrell, Tessa Lau, Stefan Nusser, Eric Wilcox, and Michael Muller. Socially augmenting employee profiles with people-tagging. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 91–100. ACM, 2007. — Cited on page 18.

[32] Hugo Fuks, Alberto B Raposo, Marco A Gerosa, and Carlos JP Lucena. Applying the 3c model to groupware development. *International Journal of Cooperative Information Systems*, 2005. — Cited on page 14.

[33] D Furniss. *Codifying distributed cognition: a case study of emergency medical dispatch.* PhD thesis, UCL (University College London), 2004. — Cited on page 32.

[34] Elihu M Gerson. Reach, bracket, and the limits of rationalized coordination: Some challenges for cscw. In *Resources, Co-Evolution and Artifacts*, pages 193–220. Springer, 2008. — Cited on page 36.

[35] Werner Geyer, Casey Dugan, Joan DiMicco, David R Millen, Beth Brownholtz, and Michael Muller. Use and reuse of shared lists as a social content type. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1545–1554. ACM, 2008. — Cited on page 18.

[36] Oliver Günther, Hanna Krasnova, Dirk Riehle, and Valentin Schöndienst. Modeling microblogging adoption in the enterprise. In *AMCIS*, page 544, 2009. — Cited on page 19.

[37] Carl Gutwin, Reagan Penner, and Kevin Schneider. Group awareness in distributed software development. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 72–81. ACM, 2004. — Cited on pages 2, 3, and 17.

[38] Christine A Halverson. Activity theory and distributed cognition: Or what does cscw need to do with theories? *Computer Supported Cooperative Work (CSCW)*, 11(1-2):243–267, 2002. — Cited on page 32.

[39] Mark Handel and James D Herbsleb. What is chat doing in the workplace? In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pages 1–10. ACM, 2002. — Cited on page 18.

[40] Hans-Jörg Happel. Social search and need-driven knowledge sharing in wikis with woogle. In *Proceedings of the 5th international Symposium on Wikis and Open Collaboration*, page 13. ACM, 2009. — Cited on page 19.

[41] James D Herbsleb. Global software engineering: The future of socio-technical coordination. In *2007 Future of Software Engineering*, pages 188–198. IEEE Computer Society, 2007. — Cited on pages 1, 15, and 16.

[42] James D Herbsleb and Deependra Moitra. Global software development. *Software, IEEE*, 18(2):16–20, 2001. — Cited on pages 14 and 15.

[43] Christine Hine. Virtual ethnography: Modes, varieties, affordances. *The SAGE handbook of online research methods*, pages 257–270, 2008. — Cited on page 29.

[44] Geert Hofstede and Michael H Bond. Hofstede's culture dimensions an independent validation using rokeach's value survey. *Journal of cross-cultural psychology*, 15(4):417–433, 1984. — Cited on page 52.

[45] James Hollan, Edwin Hutchins, and David Kirsh. Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 7(2):174–196, 2000. — Cited on page 32.

[46] Helena Holmström, Brian Fitzgerald, Pär J Ågerfalk, and Eoin Ó Conchúir. Agile practices reduce distance in global software development. *Information Systems Management*, 23(3):7–18, 2006. — Cited on page 15.

[47] Lester J Holtzblatt, Laurie E Damianos, and Daniel Weiss. Factors impeding wiki use in the enterprise: a case study. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*, pages 4661–4676. ACM, 2010. — Cited on page 19.

[48] Edwin Hutchins. *Cognition in the Wild*, volume 262082314. MIT press Cambridge, MA, 1995. — Cited on page 32.

[49] Hyun-Gyung Im, JoAnne Yates, and Wanda Orlikowski. Temporal coordination through communication: using genres in a virtual start-up organization. *Information Technology & People*, 18(2):89–119, 2005. — Cited on page 34.

[50] Sirkka L Jarvenpaa and Dorothy E Leidner. Communication and trust in global virtual teams. *Journal of Computer-Mediated Communication*, 3(4):0–0, 1998. — Cited on page 51.

[51] Brigitte Jordan and Austin Henderson. Interaction analysis: Foundations and practice. *The journal of the learning sciences*, 4(1):39–103, 1995. — Cited on pages 5, 26, and 28.

[52] Andreas M Kaplan and Michael Haenlein. Users of the world, unite! the challenges and opportunities of social media. *Business horizons*, 53(1):59–68, 2010. — Cited on pages 2 and 17.

[53] Karlheinz Kautz and Jacob Nørbjerg. Persistent problems in information systems development. the case of the world wide web. In *ECIS*, pages 919–926, 2003. — Cited on page 30.

[54] Eric Knauss, Olesia Brill, Ingo Kitzmann, and Thomas Flohr. Smartwiki: Support for high-quality requirements engineering in a collaborative setting. In *Wikis for Software Engineering, 2009. WIKIS4SE'09. ICSE Workshop on*, pages 25–35. IEEE, 2009. — Cited on pages 17, 41, and 43.

[55] Michael Koch. Cscw and enterprise 2.0–towards an integrated perspective. *21th Bled eConference, eCollaboration: Overcoming Boundaries Through Multi-Channel Interaction*, 2008. — Cited on page 16.

[56] Miia Kosonen, Kaisa Henttonen, and Hanna-Kaisa Ellonen. Weblogs and internal communication in a corporate environment: a case from the ict industry. *International Journal of Knowledge and Learning*, 3(4):437–449, 2007. — Cited on page 19.

[57] Filippo Lanubile. Collaboration in distributed software development. In *Software Engineering*. 2009. — Cited on pages 14 and 15.

[58] Pablo Lapegna. Ethnographers of the world united? current debates on the ethnographic study of “globalization”,“. *Journal of World-System Research*, 15(1):3–24, 2009. — Cited on page 29.

[59] Thomas D LaToza, Gina Venolia, and Robert DeLine. Maintaining mental models: a study of developer work habits. In *Proceedings of the 28th international conference on Software engineering*, pages 492–501. ACM, 2006. — Cited on pages 17, 41, and 43.

[60] Lucas Layman, Laurie Williams, Daniela Damian, and Hynek Bures. Essential communication practices for extreme programming in a global software development team. *Information and software technology*, 48(9):781–794, 2006. — Cited on page 15.

[61] Charlotte P Lee. Boundary negotiating artifacts: unbinding the routine of boundary objects and embracing chaos in collaborative work. *Computer Supported Cooperative Work (CSCW)*, 16(3):307–339, 2007. — Cited on page 31.

[62] Ann Majchrzak, Christian Wagner, and Dave Yates. Corporate wiki users: results of a survey. In *Proceedings of the 2006 international symposium on Wikis*, pages 99–104. ACM, 2006. — Cited on page 3.

[63] Thomas W. Malone and Kevin Crowston. The interdisciplinary study of coordination. *Computing Surveys*, 26(1):87–119, 1994. — Cited on pages 15 and 16.

[64] Christina Manteli, Hans van Vliet, and Bart van den Hooff. Adopting a social network perspective in global software development. In *International Conference on Global Software Engineering, 2012.* — Cited on page 14.

[65] George E Marcus. Ethnography in/of the world system: the emergence of multi-sited ethnography. *Annual review of anthropology*, pages 95–117, 1995. — Cited on page 29.

[66] George E Marcus. *Ethnography through thick and thin.* Princeton University Press, 1998. — Cited on page 29.

[67] Carolyn R Miller. Genre as social action. *Quarterly journal of speech*, 70(2):151–167, 1984. — Cited on page 34.

[68] Sunila Modi, Pamela Abbott, and Steve Counsell. Negotiating common ground in distributed agile development: A case study perspective. In *International Conference on Global Software Engineering, 2013.* — Cited on page 14.

[69] Davide Nicolini, Silvia Gherardi, and Dvora Yanow. *Knowing in organizations: A practice-based approach.* ME Sharpe, 2003. — Cited on pages 5 and 30.

[70] Peter Axel Nielsen and Jacob Nørbjerg. Assessing software processes: low maturity or sensible practice. *Scandinavian Journal of Information Systems*, 13(1):5, 2001. — Cited on page 30.

[71] T. Niinimaki and C. Lassenius. Experiences of instant messaging in global software development projects: A multiple case study. In *Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on*, pages 55 –64, aug. 2008. — Cited on pages 2, 17, 41, and 43.

[72] Wanda J Orlikowski and JoAnne Yates. Genre repertoire: The structuring of communicative practices in organizations. *Administrative science quarterly*, pages 541–574, 1994. — Cited on pages 34 and 44.

[73] David J Pauleen and Pak Yoong. Facilitating virtual team relationships via internet and conventional communication channels. *Internet Research*, 11(3):190–202, 2001. — Cited on page 18.

[74] Ammy Jiranida Phuwanartnurak. Did you put it on the wiki?: information sharing through wikis in interdisciplinary design collaboration. In *Proceedings of the 27th ACM international conference on Design of communication*, pages 273–280. ACM, 2009. — Cited on page 19.

[75] Shih-Ming Pi, Yi-Chih Liu, Tsang-Yao Chen, and Shih-Hua Li. The influence of instant messaging usage behavior on organizational communication satisfaction. In *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, pages 449–449. IEEE, 2008. — Cited on page 18.

[76] Lene Pries-Heje. Four different paradigms for process design when implementing standard enterprise systems. *CONFENIS 2010*, 2010. — Cited on page 36.

[77] Rafael Prikladnicki, Alexander Boden, Gabriela Avram, CleidsonR.B. Souza, and Volker Wulf. Data collection in global software engineering research: learning from past experience. *Empirical Software Engineering*, pages 1–35, 2013. — Cited on pages 25 and 27.

[78] Julia Prior, Toni Robertson, and John Leaney. Situated software development: Work practice and infrastructure are mutually constitutive. In *Software Engineering, 2008. ASWEC 2008. 19th Australian Conference on*, pages 160–169. IEEE, 2008. — Cited on page 30.

[79] Robin Privman and Starr Roxanne Hiltz. Whose (partially distributed) team are you on? interviews about" us vs. them" in corporate settings. 2008. — Cited on page 19.

[80] Anabel Quan-Haase, Joseph Cothrel, and Barry Wellman. Instant messaging for collaboration: A case study of a high-tech firm. *Journal of Computer-Mediated Communication*, 10(4):00–00, 2005. — Cited on page 19.

[81] Alexander Richter and Kai Riemer. Corporate social networking sites–modes of use and appropriation through co-evolution. In *20th Australasian Conference on Information Systems*, 2009. — Cited on page 18.

[82] Colin Robson. *Real world research: a resource for social scientists and practitioner-researchers*, volume 2. Blackwell Oxford, 2002. — Cited on pages 24, 25, 26, and 50.

[83] Kari Rönkkö, Yvonne Dittrich, and Dave Randall. When plans do not work out: How plans are used in software development projects. *Computer Supported Cooperative Work (CSCW)*, 14(5):433–468, 2005. — Cited on page 2.

[84] Kjeld Schmidt and Carla Simonee. Coordination mechanisms: Towards a conceptual foundation of cscw systems design. *Computer Supported Cooperative Work (CSCW)*, 5(2):155–200, 1996. — Cited on pages 2, 16, 19, 31, 35, 37, and 44.

[85] Carolyn B. Seaman. Qualitative methods in empirical studies of software engineering. *Software Engineering, IEEE Transactions on*, 25(4):557–572, 1999. — Cited on pages 23, 25, and 27.

[86] Anders Sigfridsson. The purposeful adaptation of practice: an empirical study of distributed software development. 2010. — Cited on page 31.

[87] Anders Sigfridsson and Anne Sheehan. On qualitative methodologies and dispersed communities: Reflections on the process of investigating an open source community. *Information and Software Technology*, 53(9):981–993, 2011. — Cited on page 30.

[88] Susan Elliott Sim, Janice Singer, and Margaret-Anne Storey. Beg, borrow, or steal: Using multidisciplinary approaches in empirical software engineering research. *Empirical Software Engineering*, 6(1):85–93, 2001. — Cited on page 23.

[89] Janice Singer, T Lethbridge, Norman Vinson, N Anquetil, et al. An examination of software engineering work practices. 1997. — Cited on page 30.

[90] Susan Leigh Star and James R Griesemer. Institutional ecology,translations' and boundary objects: Amateurs and professionals in berkeley's museum of vertebrate zoology, 1907-39. *Social studies of science*, 19(3):387–420, 1989. — Cited on page 31.

[91] Charles Steinfield, Joan M DiMicco, Nicole B Ellison, and Cliff Lampe. Bowling online: social networking and social capital within the organization. In *Proceedings of the fourth international conference on Communities and technologies*, pages 245–254. ACM, 2009. — Cited on page 18.

[92] Igor Steinmacher, AnaPaula Chaves, and Marco Aurelio Gerosa. Awareness Support in Distributed Software Development: A Systematic Review and Mapping of the Literature. *Computer Supported Cooperative Work (CSCW)*, 2013. — Cited on page 14.

[93] Anselm Strauss. The articulation of project work: An organizational process. *The Sociological Quarterly*, 29(2):163–178, 1988. — Cited on pages 31 and 36.

[94] Diane E Strode, Sid L Huff, Beverley Hope, and Sebastian Link. Coordination in co-located agile software development projects. *Journal of Systems and Software*, 85(6):1222–1238, 2012. — Cited on page 16.

[95] Lucille Alice Suchman. *Plans and situated actions: the problem of human-machine communication.* Cambridge university press, 1987. — Cited on pages 2, 5, 30, 45, and 50.

[96] John C Tang. Approaching and leave-taking: Negotiating contact in computer-mediated communication. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 14(1):5, 2007. — Cited on page 19.

[97] Brendan Tansey and Eleni Stroulia. Annoki: a mediawiki-based collaboration platform. In *Proceedings of the 1st Workshop on Web 2.0 for Software Engineering*, pages 31–36. ACM, 2010. — Cited on page 19.

[98] M Rita Thissen, Jean M Page, Madhavi C Bharathi, and Toyia L Austin. Communication tools for distributed software development teams. In *Proceedings of the 2007 ACM SIGMIS CPR conference on Computer personnel research: The global information technology workforce*, pages 28–35. ACM, 2007. — Cited on pages 17 and 19.

[99] Christoph Treude and M-A Storey. How tagging helps bridge the gap between social and technical aspects in software development. In *Software Engineering, 2009. ICSE 2009. IEEE 31st International Conference on*, pages 12–22. IEEE, 2009. — Cited on pages 2 and 18.

[100] Thea Turner, Pernilla Qvarfordt, Jacob T Biehl, Gene Golovchinsky, and Maribeth Back. Exploring the workplace communication ecology. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 841–850. ACM, 2010. — Cited on page 19.

[101] Jim Whitehead. Collaboration in software engineering: A roadmap. In *Future of Software Engineering, 2007. FOSE'07*, pages 214–225. IEEE, 2007. — Cited on page 16.

[102] Ludwig Wittgenstein. *Philosophical investigations.* Oxford: Blackwell, 1953. — Cited on page 36.

[103] Mika Yasuoka. *Bridging and breakdowns.* PhD thesis, Copenhagen Business School, 2009. — Cited on page 31.

[104] JoAnne Yates and Wanda J Orlikowski. Genres of organizational communication: A structurational approach to studying communication and media. *Academy of management review*, 17(2):299–326, 1992. — Cited on pages 33, 34, and 36.

[105] Jun Zhang, Yan Qu, Jane Cody, and Yulingling Wu. A case study of micro-blogging in the enterprise: use, value, and related issues. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 123–132. ACM, 2010. — Cited on pages 18 and 41.

[106] Dejin Zhao and Mary Beth Rosson. How and why people twitter: the role that micro-blogging plays in informal communication at work. In *Proceedings of the ACM 2009 international conference on Supporting group work*, pages 243–252. ACM, 2009. — Cited on page 19.

Empirical Studies on the Use of Social Software in Global Software Development - a Systematic Mapping Study

Rosalba Giuffrida, Yvonne Dittrich

IT University of Copenhagen - Denmark
rogi@itu.dk, ydi@itu.dk

Abstract

Background: In Global Software Development (GSD), informal communication and knowledge sharing play an important role. Social Software (SoSo) has the potential to support and foster this key responsibility. Research on the use of SoSo in GSD is still at an early stage: although a number of empirical studies on the usage of SoSo are available in related fields, there exists no comprehensive overview of what has been investigated to date across them.
Objective: The aim of this review is to map empirical studies on the usage of SoSo in Software Engineering projects and in distributed teams, and to highlight the findings of research works which could prove to be beneficial for GSD researchers and practitioners.
Method: A Systematic Mapping Study is conducted using a broad search string that allows identifying a variety of studies which can be beneficial for GSD. Papers have been retrieved through a combination of automatic search and *snowballing*, hence a wide quantitative map of the research area is provided. Additionally, text extracts from the studies are qualitatively synthesised to investigate benefits and challenges of the use of SoSo.
Results: SoSo is reported as being chiefly used as a support for collaborative work, fostering awareness, knowledge management and coordination among team members. Contrary to the evident high importance of the social aspects offered by SoSo, socialization is not the most important usage reported.
Conclusions: This review reports how SoSo is used in GSD and how it is capable of supporting GSD teams. Four emerging themes in global software engineering were identified: the appropriation and development of usage structures; understanding how an ecology of communication channels and tools are used by teams; the role played by SoSo either as a subtext or as an explicit goal; and finally, the surprising low percentage of observational studies.

Keywords: systematic mapping study, global software development, distributed teams, social media, social software, computer-supported cooperative work

1. Introduction

Global Software Development (GSD) is increasingly becoming a common practice in the software industry [6]. Organizations establish global software projects, which are scattered all round the globe, involving multiple teams located at different sites. GSD permits to lower development costs due to salary savings, to decrease development time due to time-zone effectiveness, to reduce time to market and to access the most talented developers [4]. However, since GSD is highly geographically dispersed, accompanying challenges and numerous problems have to be dealt with, overcome and solved: teams have to deal with temporal, geographical and socio-cultural distances, resulting in major difficulties in coordination and communication [5]. Researchers have highlighted issues with division of work, inadequate communication, knowledge management, project and process management difficulties and infrastructure problems [11].

In co-located teams, planned or impromptu face-to-face meetings can easily support the development processes, complementing formal knowledge and documentation [3]. While, traditionally, main media for informal communication in distributed teams are email, phone and video conferencing system, our belief is that, nowadays, other kinds of unstructured channels, such as Social Software (SoSo) [13], can ultimately support and foster informal communication and knowledge sharing in GSD teams. Thus, information shared through SoSo, complements formal communication and documentation shared through traditional structured tools. This idea is confirmed by Black et al. [2]: in their study, SoSo is reported to be successfully used by GSD developers to communicate at work, resulting in improvement of communication with other members of the team, sharing technical information and suggesting new ideas. Different kinds of SoSo have characteristics that offer support for diverse GSD challenges. For example, Instant Messaging (IM) has been reported as a way to accelerate communication and to obtain rapid feedback [LVD06, TPBA07], while Wikis are used for managing documentation and sharing knowledge [KBKF09]. In our view, SoSo has the potential to let people collaborate easily regardless of the distance, to foster frequent informal communication, to provide awareness and to simplify information sharing through lightweight tools.

While research on SoSo in GSD is still at an early stage, there are several empirical studies on the usage of SoSo in related fields, such as Software Engineering (SE) and Computer-supported Cooperative Work (CSCW). In these areas, there is a growing interest in SoSo, and specialised workshops have recently been established, such as Social Media at Work [1], Web2SE [2] and Social Software Engineering [3].

Consequently, there arises a requirement to provide GSD researchers and practitioners with an overview of the current state of the art. Moreover, the results of a systematic mapping study would also prove to be beneficial for the CSCW and SE communities, which currently fall short of possessing a comprehensive secondary study aggregating all contributions.

Hence, we conducted a Systematic Mapping Study (SMS) to establish evidence of the quantity of existing research, and thereby to provide a broad overview of the research results, which are in line with the SMS purposes proposed by Kitchenham and Charters [14].

The organisational structure for the paper is based on the following guidelines: the next section discusses the motivation behind this research and summarises related work. Section 3 describes the research methodology, detailing the different steps carried out for conducting this SMS. The results of the mapping are presented in Section 4, and Section 5 is a detailed analysis and reporting of the qualitative findings on the benefits and challenges of using SoSo. Section 6 discusses the results, answers the research questions, highlights the contributions for GSD areas and the emerging themes, and in brief, reports the limitations of the study. Final conclusions are presented in Section 7.

2. Motivation and Background

In this section, concrete explanations are offered for the motivations and objectives behind this study, followed by a definition of SoSo, culminating with placing SoSo in the context of SE and CSCW research.

2.1. Motivations and Objectives

This Systematic Mapping Study (SMS) is part of a PhD project aimed at studying how SoSo is used GSD teams [9]. Corresponding to the main research topic, the primary research question that motivated this study and the consequent investigation was centred on endeavouring to understand how SoSo could support virtual teams in Global Software Development. Based upon an initial, non-systematic literature review it was swiftly evident that research in this area is still at an early stage. In fact, the only publications retrievable on this subject, through a manual search, were works by Niinimaki and Lassenius [NL08], by Black et al. [2] and by Thissen et al. [TPBA07]. Despite the fact that there has not been extensive research conducted on this particular topic, there is a growing interest in investigating the use of SoSo in distributed teams and in SE settings (see also Subsection 2.3). This interest persists in spite of the fact that both these areas are not yet well established and systematic reviews for further research are unavailable. Therefore, this SMS was carried out with the sole purpose of affording an extensive and detailed overview of the research area. For conducting the review, the guidelines followed were described by Petersen et al. [17] in their paper on conducting a SMS in Software Engineering.

Systematic Mapping Study (SMS) is an alternative to Systematic Literature Reviews (SLR) and is employed under circumstances when there is insufficient empirical evidence available or when the topic area is too vast for it to be feasible to conduct a SLR [17] . For this paper, a SMS was conducted primarily for two main reasons: the evidence present in the GSD field was inadequate and it would not have been practicable to conduct a SLR which extended both to distributed teams and to SE projects. This paper aims to present not only an overview of the area, but also an understanding of how SoSo is used in different settings, therefore the SMS was extended with a deeper analysis of the papers, providing qualitative analysis for complementing the quantitative outcomes of the SMS, as suggested by Petersen et al. [17].

2.2. Social Software

SoSo is described by Farkas [8] as tools that: (1) Allow people to communicate, collaborate, and to build communities online. (2) Can be shared, reused or remixed. (3) Let people learn easily from and capitalise on the behaviour and knowledge of others. SoSo include a wide variety of tools such as: Instant Messaging (IM), Blog, Microblog, Wiki, Social Networking Sites (SNS), and Social Bookmarking. Scholars and practitioners often refer to these tools also as *Social Media* or *Web 2.0 tools*. A definition of Social Media has been recently developed by Kaplan and Haenlein [13]: *A group of Internet-based applications built on the ideological and technological foundations of Web 2.0, and that allow the creation and exchange of User Generated Content (UGC)*. This definition is consistent with the one by Farkas [8]. However, in this study, these tools for the most part, are referred to as SoSo rather than as Social Media, as the media component (in the sense of broadcasting information in a publicly accessible website) is not essential for communication among team members in an industrial setting.

This difference in terminology has several implications for this study, and, as it will be further described in Subsection 3.4, several keywords have to be considered while constructing the query string, due to the lack of a common terminology established among practitioners and researchers.

2.3. Related Work

In the Computer Supported Cooperative Work (CSCW) community, scholars have now for several decades been researching support for collaboration in work groups. As Koch [15] reports, *SoSo entered this field from the sidelines, and quickly expanded*

[1] http://www.ecscw2011.org/wordpress/wp-content/uploads/2011/05/ws7_final.pdf

[2] https://sites.google.com/site/web2se/

[3] http://www1.cs.tum.edu/static/sse11/

towards support for collaboration in enterprises. However, since the interaction between CSCW and the social computing field is minimal, Koch tried to bridge the gap by identifying the core contributions of the two fields, and how they can be integrated or used to benefit each other [15].

In the Software Engineering (SE) community, there is a noticeable growing interest in what is termed Social Software Engineering. Ahmadi et al. [1] provided a survey of contributions in areas related to SoSo Engineering, such as Psychology, Mathematics and Computer Science studies. They highlighted the fact that the combination provided by these fields is required as a fundamental basis for engineering social and collaborative applications as well as the SE process deploying them [1].

Despite the growing interest in SoSo, it was difficult to come across any SLR or SMS on the use of SoSo during the pilot study for the SMS presented here. Nonetheless, there were some non-systematic reviews focussing on specific kinds of SoSo, which were obtainable. Raeth and Smolnik [18] reviewed the existing research on corporate blogging, and Ioanna et al. [16] explored the effects that wiki technology has on a variety of organisational processes. Both reviews are non-systematic and they do not describe a clear process to obtain the papers or to even include or exclude such documents. In any case, qualitative insights given by these reviews are particularly relevant for SoSo research and would need to be taken into account to extend the findings of this SMS.

3. Research method

3.1. Pilot Study

Prior to the present SMS, a pilot study has been performed to verify whether there was a need for a Systematic Mapping Study and to clarify the research questions. Suitable resources were searched and possible search terms were tested likewise.

During the initial phase of the pilot study, proceedings of the main conferences in the GSD field were analysed. Only one work [NL08] was found, confirming that research in the area is still at a very early stage. Thus, a query string sufficiently comprehensive for including research from related fields was constructed, and meta-repositories, such as Google Scholar, were queried, as a test for checking the broadness of the field. At once, it was evident that related research was spread across different communities (eg. the work from Thissen et al.[TPBA07] was found). Analysing all the proceedings of all the conferences, workshops and journals would not be so efficient. Thus, the decision of performing this SMS was made, informed by the fact that relevant papers were retrievable in traditional repositories, such as Springer Link, Science Direct, The Institute of Electrical and Electronics Engineers (IEEE), etc.

The pilot study was ultimately used to define a formal protocol that guided the research objectives and defined how the mapping study would have to be performed, defining research questions, and planning how the sources and studies would be utilised to answer these questions. The SMS presented here has been conducted according to this protocol. Incremental reviews to the protocol were performed to update the protocol, based upon new information collected as the study progressed.

Next Subsections outline the protocol in detail and describes how the study has been conducted in accordance with this protocol.

3.2. Scope of the Study

The research questions were framed by the following PICO criteria [14]:

- **Population:** Distributed teams and SE teams.
- **Intervention:** Empirical studies involving the use of SoSo in distributed teams and in SE projects.
- **Comparison:** No comparisons takes place for the purpose of this study.
- **Outcomes:** Usage of SoSo in distributed teams, quantitative, as well as qualitative results (benefits and challenges).

3.3. Research questions

To examine the evidence on the state of research on SoSo in distributed teams, the following research questions were considered:

> **RQ1.** What are the fora in which research on SoSo in distributed teams and software engineering has been published to date?
> **RQ2.** How is SoSo reported to be used in distributed teams and in SE?
> **RQ3.** What are the benefits and the challenges of using SoSo in distributed teams and in SE?

3.4. Search String

As presented below, the search string was defined as a combination of two set of keywords focussing on SoSo and on the context.

{Social Software} **AND** {context}

SoSo as already noted in Subsection 2.2, comprises several terms used to refer to SoSo by practitioners and researchers:

> ”social media”, ”social software”, ”web 2.0”, ”user generated media”, ”user generated content”, ”instant messaging”, wiki, ”social network”, ”social bookmark”, blog, microblog, ”social tagging”, facebook, twitter

Context is defined through diverse terminology used to identify: (a) teams which are not co-located; (b) SE and its main phases; and (c) known purposes for using SoSo [4].

a) Teams which are not co-located would also be referred to as: ”virtual team”, ”virtual teams”, ”global team”, ”global teams”, ”distributed team”, ”distributed teams”

[4] A possible further continuation of this study might consider to add the purposes found during this review for (c).

b) SE and its main phases would allude to: "software engineering", "software development", "requirement engineering", "software quality", "requirement specification", "software testing", "software architecture", "quality assurance"

c) Known purposes for using SoSo would be: "collaborative work", "distributed work", "knowledge sharing", "informal communication"

Terms in each set have been combined using the Boolean operator OR. Hence, the final search query would read as:

> ("social media" OR "social software" OR "web 2.0" OR "user generated media" OR "user generated content" OR microblog OR blog OR "instant messaging" OR "social network" OR facebook OR twitter OR "social bookmark" OR "social tagging" OR wiki) **AND** ("virtual team" OR "virtual teams" OR "global team" OR "global teams" OR "distributed team" OR "distributed teams" OR "software engineering" OR "software development" OR "requirement engineering" OR "software quality" OR "requirement specification" OR "software testing" OR "software architecture" OR "quality assurance" OR "collaborative work" OR "distributed work" OR "knowledge sharing" OR "informal communication")

The Search String is broad enough to assure that different possible usage of SoSo that can be related to GSD are included.

3.5. Search Strategy

The Search Strategy is depicted in Figure 1 and comprises two main stages: Automatic Search to identify primary studies and Manual Search on the references of primary studies to extend the review and thereby, include additional studies. During the Automatic Search stage (Figure 1 - Auto), electronic databases were queried using the search string which was built as reported in Subsection 3.4. The databases searched were:

- Association for Computing Machinery (ACM)
- Association for Information Systems Electronic Library (AISeL)
- The Institute of Electrical and Electronics Engineers (IEEE)
- Science Direct
- Springer Link

Once the initial data was obtained, using the search string in the selected data sources, the papers were analysed, in order to consider their relevance to research questions as well as based upon the inclusion and exclusion criteria (described in Subsection 3.6). Mendeley[5], a desktop and web program for managing and sharing research papers, has been used for storing all citations. Initially, the exclusion criteria have been applied only to the Title, then both exclusion and inclusion criteria were applied to the Abstract and to the Conclusion and the classification scheme was developed (see Subsection 3.7; the classification schema is reported in Appendix A). When Abstracts were recognised to be poor in quality, the Introduction and Conclusion sections of the papers were also analysed and, if necessary, then the entire paper was scrutinised and analysed. The citations were then imported into a spreadsheet that was also used for the data extraction phase (see Section 3.7).

A Manual Search stage (Figure 1 - Manual) has been performed as an extension of the initial study: all the references of the primary studies were reviewed and the exclusion criteria have been applied to the Title. The process of pursuing references of references is known as *snowballing* [10]. The new references obtained by the snowballing process have been imported in Mendeley and duplicates have been automatically removed. Abstracts and Conclusions of the potential secondary studies were analysed and exclusion and inclusion criteria have been applied. When necessary, the entire paper has been retrieved and analysed. The ulterior studies obtained from the Manual Search stage have been added to the primary studies, which were the result of the Automatic Search Stage, producing the final set of studies.

3.6. Inclusion and Exclusion Criteria

The key criterion needed by a publication to be included was the relevance of the paper towards the main research topic covered by this study, i.e. describing empirical research on SoSo in distributed teams or in SE projects, mainly for supporting informal communication, knowledge sharing, collaboration and coordination in distributed teams. Pure discussion/opinion papers and literature reviews have been excluded, as well as papers about trends in SoSo - the study only utilises empirical cases about the use of SoSo, in terms of both quantitative and qualitative studies. Additionally, papers about the use of SoSo in e-Health, e-Government, Business, Marketing and e-Learning have been excluded. However, papers reporting on student projects have been retained for use, if the scenario and the findings resembled a work environment. Learning outcomes or educational findings possessed no special interest for the purposes of this study, but if useful practices for CSCW/SE were identified, then those studies were definitely included. Articles on how to develop SoSo applications have also not been included in the study. Papers from the CHI community related to topics covering, for example, interface design, have been included only when effects on collaboration were analysed (i.e. through an evaluation of the system). Papers focussed on interface issues have been excluded, since the primary interest for the study is how these systems are used in real cases/experimental settings. Studies about usage of SoSo in teams have been included. However, any study analysing the effects of the usage of SoSo on individuals has not even been considered for this paper. Finally, only journal, conference and workshop papers larger than six pages have been included, excluding all position papers and CHI "work in progress" articles.

When more papers reporting the same study were retrieved, only the more complete work was included in the review. If

[5] http://www.mendeley.com/

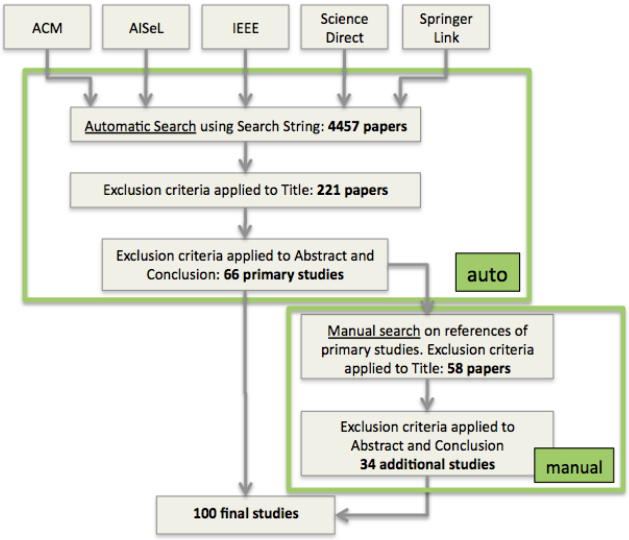

ACM
AISeL
IEEE
Science Direct
Springer Link
Automatic Search using Search String: 4457 papers
Exclusion criteria applied to Title: 221 papers
Exclusion criteria applied to Abstract and Conclusion: 66 primary studies
auto
Manual search on references of primary studies. Exclusion criteria applied to Title: 58 papers
Exclusion criteria applied to Abstract and Conclusion 34 additional studies
manual
100 final studies

Figure 1: Search Strategy

a paper explicitly stated that it was a continuation of previous works, the most recent paper was inserted in the review and the initial study eliminated. For example, the first study of Hubbub [12] has been excluded and only the more recent study has been included [IWW+02]. However, in the case of [DMG+08] and [DGM+09], multiple publications of the same work have been utilised, while highlighting different perspectives and providing complementary findings.

3.7. *Data Extraction and Synthesis*

The classification scheme was developed while applying exclusion and inclusion criteria to Abstract and Conclusion of potential primary studies during the Automatic Search stage (see Subsection 3.5). This activity is called *keywording* [17]: it is a method to reduce the time needed for defining the classification scheme and, in addition, it ensures that the scheme takes existing studies into account. Through this method, a set of categories was defined which was representative of the underlying population. The classification schema developed is reported in Appendix A.

The items in the form were selected in alignment with the objectives of this study, and aimed at enabling the authors to answer the research questions by analysing the extracted data. The data required for the analysis have been extracted by exploring the full-text of each included paper. A spreadsheet was used for data extraction and collection. Text extracts from the papers have been collected to include discursive analysis of benefits and challenges of the use of SoSo.

Once the data from the studies were recorded, a quantitative and qualitative analysis was performed. From this analysis, characteristics and properties were identified according to the objectives and research questions proposed. The next section presents the quantitative collected data.

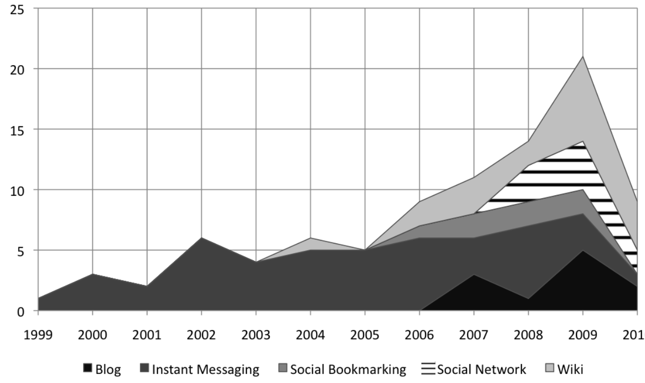


Figure 2: Distribution of papers about each SoSo over years.

Table 1: Amount of studies for each SoSo

Social Software	Number
Instant Messaging	45
Wiki	19
Blog	11
Social Network	9
More than one SoSo	9
Social Bookmarking	7

4. Results

From the first stage of the Search (Automatic Search Stage), 4457 papers were retrieved [6]. The Automatic Search Stage was performed the 22nd July 2010, so the initial set of 4457 primary studies is limited to the papers that have been already published at that date. 66 primary studies were selected after applying the inclusion and exclusion criteria to the initial set of papers (Figure 1 - Auto) and 34 additional studies have been identified from the Manual Search stage (Figure 1 - Manual). The resulting 100 papers [7] constitute the final set of studies analysed in this article. The complete reference list is available in Appendix B.

The distribution of the 100 papers over the years is reported in Figure 2, and it very evidently demonstrates a growing interest in this topic. The various types of Social Software considered in the papers are highlighted in the graph. It is however to be noted that the search was conducted in July 2010, therefore all papers published in 2010 are not included in the Mapping Study. For this reason, there is a decrease in the number of papers in 2010, which would in normal circumstances have followed the growth trend of the previous years.

4.1. *Overview of Studies*

As reported in Table 1, extensive research has been conducted over a period of time on Instant Messaging (IM), since

[6] The BibTeX file of whole set of the initial studies (4457 papers) is available at the address: `www.itu.dk/people/rogi/SMSinGSDinitialSet.bib`

[7] The BibTeX file of the 100 final references is available at the address: `www.itu.dk/people/rogi/SMSinGSDfinal.bib`

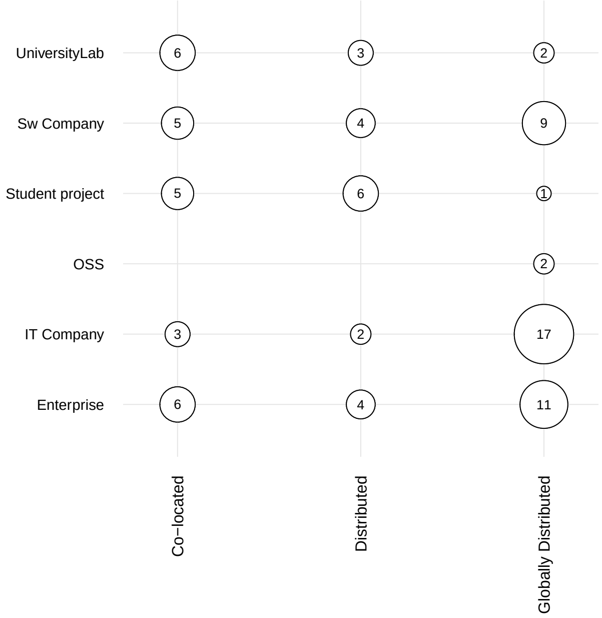


Figure 3: Population studied

it appears to be the most popular form of SoSo, allowing instantaneous effective and efficient communication. Research on various other types of SoSo is more recent, owing to their late spread in mainstream usage. It is interesting to note that research tends to follow the mainstream popularity of different tools. For example, the rising interest of Google users in Wikis and Blog dates back to between 2004 and 2006 (comparing the data from Google trends on keywords "Wiki" and "Blog" [8]). Data collected for this study corroborates the fact that it is between the years 2004 and 2006 that actual research started on these tools, in spite of the fact that they were available since early 2000. Wiki and Blog, like other SoSo, were adopted by companies and therefore seriously researched in academia, only after they became mainstream tools of SoSo. Due to the growing popularity of SoSo, we believe that the current spread and usage of SoSo in organizations is even bigger than what appears from the data reported in this SMS, leading to the fact that it is imperative to conduct further research in this field.

4.2. Population Studied

Empirical studies performed in different organisational settings were collected for this study. Figure 3 presents the population researched in the papers with respect to the category of organisation and type of distribution of the team. Most of the papers report studies in globally distributed settings, but additionally, some papers also share insights about the usage of SoSo in co-located work environments. Student projects and university laboratory experiments are a relevant part of the review, but the major portion of the research is conducted in industrial settings.

Table 2: Empirical Research Methods adopted

Method	Number
Quantitative	22
Qualitative	24
Mix quantitative and qualitative	54
Interview	59
Survey	43
Activity log	40
Logs Content Analysis	36
Experiment	15
Observations	11
Report own experience	8
User testing	3

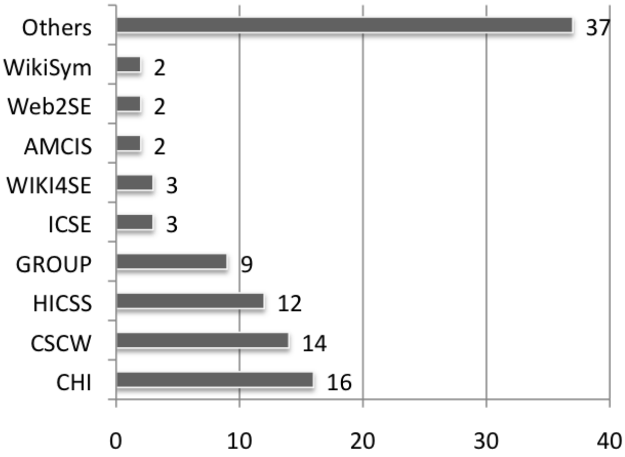


Figure 4: Main publication fora

4.3. Research Methods

Since the studies included in this review are all empirical cases, data about research methods adopted in the studies have been extracted as well. A detailed analysis of research methods is outside the scope of this article. However, from Table 2, it is evident that both quantitative and qualitative methods have been equally used. It is also apparent that both quantitative and qualitative methods have been used in combination for over half of the study, as a means of complementing each other. In many studies more than one method of data collection was adopted, favouring the use of interviews and surveys. Logs of SoSo are analysed in a quantitative way (*activity log*), but the content of the logs is often analysed as well (*logs content analysis*). Only 11 studies include observations as (part of their) methods.

4.4. Publication fora

It was expected that literature about SoSo in distributed teams and SE would be spread among different research communities and very different publication fora.

The distribution of papers in publication fora is reported in Figure 4; the large number of "Others" indicates that the reach

[8] `http://www.google.com/trends/?q=blog,+wiki`

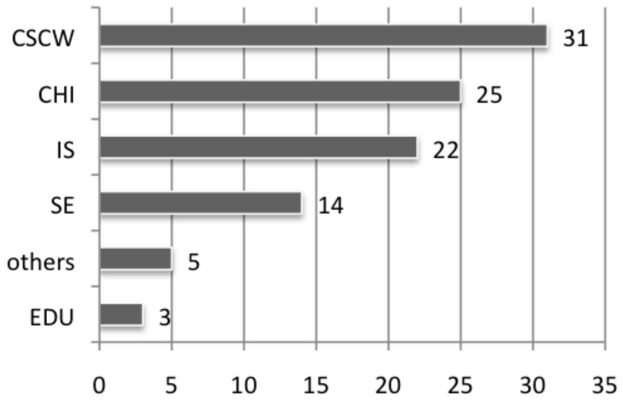


Figure 5: Communities

of the search was extensive. The 37 papers belonging to "Others" appear in 37 different fora.

As observed from the synthesis reported in Figure 5, most of the papers are published in the CSCW community, but the CHI and IS communities also provide the basis for a large proportion of studies conducted. Interestingly, there were only 14 papers available from SE venues; however, empirical studies conducted in Software Companies are bigger, as seen in Figure 3. This would mean that some SE studies are published from other venues as well.

Figure 6, illustrates how IM and Wiki have been investigated in all communities, while Social Network and Blog are mainly researched in the CHI and IS communities. The various types of SoSo have all been investigated at some point in time in SE contexts, indicating an active interest in these kinds of tools. A singularly important point to be noted is that Wikis and IM are the ones most widely researched in SE contexts and research on other kinds of SoSo is not yet widely spread in SE.

4.5. *Usage of Social Software*

More than half the papers concentrate on investigating the communication aspect of the specific SoSo. This is to be expected, in view of the fact that all varieties of SoSo can basically be regarded as communication tools. Furthermore, additional elements of the 3C (Communication, Collaboration and Coordination) model [20] are analysed in the papers, with the main focus being on collaboration, as well as exploring awareness and coordination aspects, as seen in Figure 7. Many papers included in the study concentrate on analysing the knowledge management perspective of SoSo. Interestingly, socialising is reported as an important aspect, but not as the most relevant aspect. Despite the fact that one of the chief purposes of SoSo is that it is considered as vital to support socialising, the tools are mainly used as a support for collaborative work. A detailed analysis of these themes is conducted in Section 5, reporting on how divergent aspects are investigated for each category of SoSo.

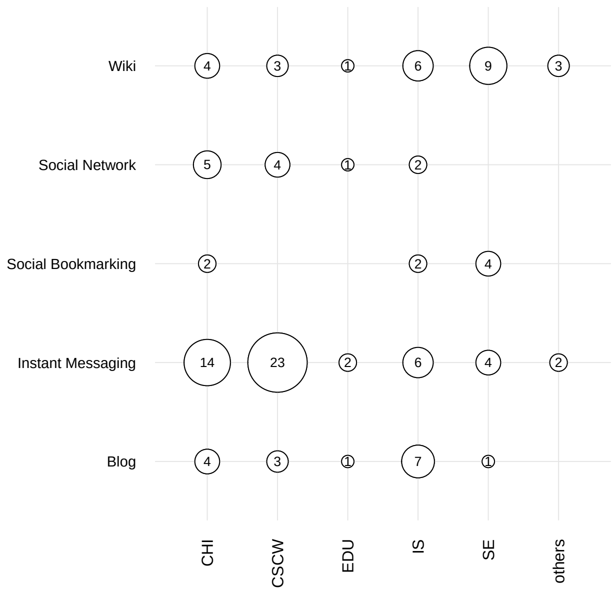


Figure 6: Distribution of SoSo for community

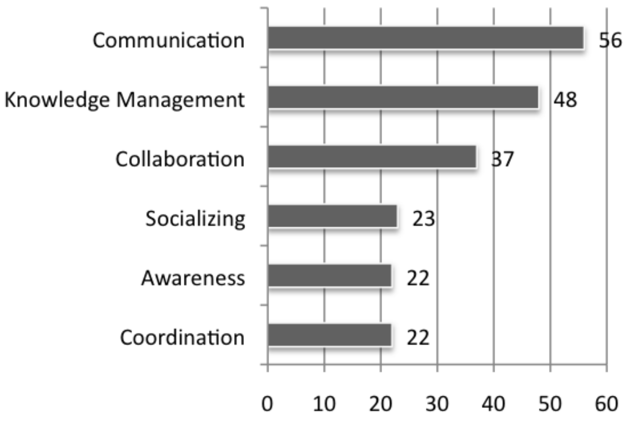


Figure 7: Themes

5. Benefits and Challenges of the Usage of SoSo

In this section, a detailed analysis of the qualitative findings of the papers included in the review is reported. The analysis has been performed reporting text extracts from the studies that identify benefits and challenges of the use of SoSo. For each category of SoSo identified, benefits of usage and challenges are summarised, both from an organizational perspective as well as from a SE standpoint. The analysis ends with papers that investigate more than one SoSo in the same empirical work.

The following subsections aim to present an in-depth and comprehensive analysis of reported usage of SoSo, provide an overview of all the studies identified for the respective SoSo in a reference table, and ultimately, to summarise benefits and challenges identified by the studies. Since each table contains all studies that investigate a specific SoSo, it follows that papers that analyse more than one SoSo are reported in more than one table.

5.1. Instant Messaging

Instant messaging (IM) is a communication tool, which offers an instantaneous transmission of text-based messages from sender to receiver. It may address point-to-point communications as well as multicast communications from one sender to many receivers (*group chat*). This review irrefutably explains that IM is one of the tools that have received the maximum share of attention from the research community to date. The reasons behind this heightened interest can be attributed to and are most probably related to its early diffusion. In fact, empirical studies on IM have been published as early as 1999 (see Figure 2).

Extensive research has been carried out on IM tools specifically developed for organisations or for research purposes. The tools identified are: Awarenex [Tan07], Babble [BKE02], Copper [NFM$^+$03], Hubbub [ESK$^+$99, IWW$^+$02], IMhere [HRS04], Loops[ESK$^+$99, EKL$^+$06], ProjectView [FKSS04], ReachOut [JSU03], Rear View Mirror [HAB$^+$02, HH02] and WebWho [SL02]. A detailed analysis of these tools is outside the scope of this review, but these studies provide contributions that can be related to IM in general.

In the following subsections, benefits and challenges identified for IM are reported; and the complete list of references of studies on IM is reported in Table 3.

5.1.1. Benefits

IM is a cost-effective technology tool suitable for both simple and complex tasks [OZDL10]. In the workplace, IM is primarily used to exchange work-related messages, for articulation work, to coordinate projects and meetings and to negotiate colleagues' availability for discussions [HH02, QC05]. IM leads to higher connectivity and new forms of collaboration [QC05, CTK05]: most collaboration via IM consists in asking questions related to specific aspects of work; IM permits asking for advice on demand [QC05] and furthermore, it facilitates knowledge sharing [OZDL10].

The primary use of workplace IM is for work discussions, and not for social purposes. Occasions that demand the use of IM for social purposes are relatively rare [IWW$^+$02, QC05, AH06]. Some authors [SL02] report that IM is also used for coordinating social activities and for keeping in touch. However, contrary to doubts that IM might reduce productivity because employees tend to use IM for personal chatting, Pi et al. [PLCL08] identified that IM can effectively reduce communication costs and improve the efficiency of communication in an enterprise. Employees use IM not only as a replacement for other communication media but also as an additional method for reaching out to others [CW05]. IM can lead to spontaneous, informal conversations between team members. It can be perceived as a "backdoor" channel that may encourage informal communication and relationship building [NWB00, PY01]. IM provides opportunities for instant and spontaneous conversations (virtual corridor talk) [CTK05]. It has been reported that chat is used for informal communication along with video: it is not an alternative merely used when audio communication is unavailable [SMH06, NL08]. It has been found that sometimes people switched from IM when they intentionally used IM to initiate a meeting, but not because IM proved inadequate for their requirements [IWW$^+$02].

IM provides close to real time communication and is often faster than email [PLCL08]. Only partial commitment on the part of the user is required [Tan07], and users are very often able to multi-task and carry on with other activities while chatting and utilising IM. IM allows also asynchronous communication, and therefore it can reduce the number of interruptions [SMH06]. Group chat tends to be less intrusive than personal IM, since it is considered less synchronous than traditional IM. Group chat provides users with a "plausible deniability" factor, and the group addressee leads to a "diffusion of responsibility" [HH02]. Moreover, IM provides awareness information though the Contact List; this is considered valuable among distributed teams: the preview of potential availability is considered a distinct advantage over the use of phones and even face-to-face interaction [Tan07].

Many articles investigate how culture affects the use of IM for collaboration [BM02, DFL08, DFL09, WFS09], and how IM can reduce (but not eliminate) cultural differences [SFN04].

In distributed software development teams IM is used to get quick answers and immediate feedback [LVD06, TPBA07]. Niinimaki and Lassenius [NL08] reported that in successful GSD projects, the use of IM is more wide-spread to facilitate multitasking and communication with numerous people simultaneously, and as a secondary channel for communication during meetings. Moreover, through IM, distributed developers maintain both a general awareness of the entire team and more detailed knowledge of people they plan to work with [GPS04]. Interestingly, in Open Source Software (OSS) the importance to record logs has been highlighted as a means for conflict resolution in the virtual community [ES03].

5.1.2. Challenges

The informal and spontaneous nature of IM is supposed to reduce communication barriers between remote collaborators; however, social distance still seems to play a role in determining employees' perception of the effectiveness of IM in working relationship development, especially when they need to maintain some formality with communication partners [CTK05]. The social settings at work, including norms, social structure and power relations, are liable to affect how people collaborate, communicate and coordinate [WYO07] and it is noticed that very frequently employees are reluctant to use IM for reporting to their superiors [CTK05]. It appears necessary to establish social conventions of IM usage [Tan07] (for e.g. in one study implicit norms dictate that IM takes priority over other media [QC05]). Critical mass represents an important factor for IM success in the workplace [CW05]. All things considered, a better understanding of the factors affecting IM communication is needed: in particular, an improved and enhanced awareness and comprehension of differences and similarities between social and workplace IM and how the two may coexist [AH06].

Table 3: Papers on Instant Messaging

Author	Title	Year
Avrahami and Hudson [AH06]	Communication Characteristics of Instant Messaging: Effects and Predictions of Interpersonal Relationships	2006
Avrahami et al. [AFH08]	IM waiting: timing and responsiveness in semi-synchronous communication	2008
Bietz [Bie08]	Effects of communication media on the interpretation of critical feedback	2008
Birnholtz et al. [BFHB05]	Grounding needs: achieving common ground via lightweight chat in large, distributed, ad-hoc groups	2005
Black and Jacobs [BJ10]	Using Web 2.0 to improve software quality	2010
Bradner and Mark [BM02]	Why distance matters: effects on cooperation, persuasion and deception	2002
Bradner et al. [BKE02]	The Adoption and Use of ' BABBLE ': A Field Study of Chat in the Workplace	2002
Cameron and Webster [CW05]	Unintended consequences of emerging communication technologies: Instant Messaging in the workplace	2005
Cho et al. [CTK05]	The Use of Instant Messaging in Working Relationship Development: A Case Study	2005
Connell et al. [CMRC01]	Effects of communication medium on interpersonal perceptions	2001
Diamant et al. [DFL08]	Where did we turn wrong?: unpacking the effect of culture and technology on attributions of team performance	2008
Diamantet al. [DFL09]	Collaborating across cultural and technological boundaries: team culture and information use in a map navigation task	2009
Elliott and Scacchi [ES03]	Free software developers as an occupational community: resolving conflicts and fostering collaboration	2003
Erickson et al. [ESK+99]	Socially Translucent Systems: Social Proxies, Persistent Conversation, and the Design of Babble	1999
Erickson et al. [EKL+06]	A persistent chat space for work groups: the design, evaluation and deployment of loops	2006
Fussellet al. [FKSS04]	Effects of instant messaging on the management of multiple project trajectories	2004
Gergle et al. [GMKF04]	Persistence Matters: Making the Most of Chat in Tightly-Coupled Work	2004
Gotel et al. [GKSN08]	Working Across Borders: Overcoming Culturally-Based Technology Challenges in Student Global Software Development	2008
Gotel et al. [GKP+09]	Evolving an infrastructure for student global software development projects: lessons for industry	2009
Gutwin et al. [GPS04]	Group awareness in distributed software development	2004
Handel and Herbsleb [HH02]	What is chat doing in the workplace?	2002
Herbsleb et al. [HAB+02]	Introducing instant messaging and chat in the workplace	2002
Huanget al. [HRS04]	IM here: public instant messaging on large, shared displays for workgroup interactions	2004
Hung et al. [HDKC08]	Reexamining Media Capacity Theories Using Workplace Instant Messaging	2008
Isaacs et al. [IWW+02]	The character, functions, and styles of instant messaging in the workplace	2002
Jacovi et al.[JSU03]	Why do we ReachOut?: functions of a semi-persistent peer support tool[18] (abstract only)	2003
LaToza et al. [LVD06]	Maintaining mental models: a study of developer work habits	2006
Linebarger et al. [LSEP05]	Benefits of synchronous collaboration support for an application-centered analysis team working on complex problems: a case study	2005
Nardi et al.[NWB00]	Interaction and outeraction: instant messaging in action	2000
Natsu et al. [NFM+03]	Distributed pair programming on the Web	2003
Niinimaki and Lassenius [NL08]	Experiences of Instant Messaging in Global Software Development Projects: A Multiple Case Study	2008
O'Neill and Martin [OM03]	Text chat in action	2003
Ou et al. [OZDL10]	Can instant messaging empower teams at work?	2010
Pauleen and Yoong [PY01]	Facilitating Virtual Team Relationships via Internet and Conventional Communication Channels	2001
Perttunen et al. [PRKT05]	Experiments on mobile context-aware instant messaging	2005
Pi et al. [PLCL08]	The Influence of Instant Messaging Usage Behavior on Organizational Communication Satisfaction	2008

Continued on next page

Table 3 – *Continued from previous page*

Author	**Title**	**Year**
Privman and Hiltz [PH08]	Whose (Partially Distributed) Team Are You On? Interviews About ”Us vs. Them” in Corporate Settings	2008
Quan-Haase [QC05]	Instant Messaging for Collaboration: A Case Study of a High-Tech Firm	2005
Rennecker et al. [RaDH06]	Reconstructing the Stage: The Use of Instant Messaging to Restructure Meeting Boundaries	2006
Scholl et al. [SMH06]	A comparison of chat and audio in media rich environments	2006
Segerstad and Ljungstrand [SL02]	Instant messaging with WebWho	2002
Setlock et al. [SFN04]	Taking it out of context: collaborating within and across cultures in face-to-face settings and via instant messaging	2004
Smith et al. [SCB00]	Conversation Trees and Threaded Chats	2000
Tang [Tan07]	Approaching and leave-taking: Negotiating contact in computer-mediated communication	2007
Thissen et al. [TPBA07]	Communication tools for distributed software development teams	2007
Turner et al. [TQB$^+$10]	Exploring the workplace communication ecology	2010
Wang et al. [WFS09]	Cultural difference and adaptation of communication styles in computer-mediated group brainstorming	2009
Weisz et al. [WEK06]	Synchronous broadcast messaging: the use of ICT	2006
Woerner et al. [WYO07]	Conversational Coherence in Instant Messaging and Getting Work Done	2007
Young et al. [YML$^+$00]	Student Group Working Across Universities: A Case Study in Software Engineering	2000

Table 4: Papers on Blog and Microblog

Author	Title	Year
Barnes et al. [BBKS10]	Towards an Understanding of Social Software: The Case of Arinia	2010
Black and Jacobs [BJ10]	Using Web 2.0 to improve software quality	2010
Böhringer and Richter [BR09]	Adopting Enterprise 2.0: A Case Study on Microblogging	2009
Brzozowski [Brz09]	WaterCooler: exploring an organization through enterprise social media	2009
Brzozowski et al. [BSH09]	Effects of feedback and peer pressure on contributions to enterprise social media	2009
Efimova and Grudin [EG07]	Crossing Boundaries: A Case Study of Employee Blogging	2007
Gotel [GKSN08]	Working Across Borders: Overcoming Culturally-Based Technology Challenges in Student Global Software Development	2008
Günther et al. [GKRS09]	Modeling Microblogging Adoption in the Enterprise	2009
Jackson et al. [JYO07]	Corporate Blogging: Building community through persistent digital talk	2007
Kosonen et al. [KHE07]	Weblogs and internal communication in a corporate environment: a case from the ICT industry	2007
Stocker and Tochtermann [ST08]	Investigating Weblogs in Small and Medium Enterprises: An Exploratory Case Study	2008
Turner et al. [TQB$^+$10]	Exploring the workplace communication ecology	2010
Wattal et al. [WRM09]	Employee Adoption of Corporate Blogs: A Quantitative Analysis	2009
Yardi et al. [YGB09]	Blogging at work and the corporate attention economy	2009
Zhang et al. [ZQCW10]	A case study of micro-blogging in the enterprise: use, value, and related issues	2010
Zhao and Rosson [ZR09]	How and why people Twitter: the role that micro-blogging plays in informal communication at work	2009

5.2. Blog and Microblog

A Blog is a website that contains an online personal journal with reflections, comments, and often hyperlinks provided by the writer. Microblog is a broadcast medium in the form of blogging. Microblogging allows users to exchange small elements of content such as short sentences, individual images, or video links. Typically, the content of a Microblog is smaller than that of a Blog. Contrary to a Wiki (see Paragraph 5.3), where the opinion of the individual user disappears in favour of a more impartial collective judgement, a weblog is author-centered, expressing the author's subjective point of view [ST08].

5.2.1. Benefits

Blog is considered an effective means for sharing organisational knowledge in an informal manner [KHE07]; the purpose of internal blogging is to facilitate bottom-up knowledge sharing and to foster social relations among employees [YGB09, JYO07]. Weblogs make implicit knowledge explicit in an unsolicited, self-organised manner [ST08]. A weblog can accelerate information flow, increase productivity, improve reputation; it is an arena for negotiation and interplay between personal and corporate interests [EG07]. When encouraged, employee weblogs can change how work is organised and how authority is distributed by fostering direct communication across organisational boundaries[EG07].

Benefits for users are social and informational - blogging enhances people's sense of organisational citizenship [JYO07] and it can provide a space to share passion for work, to document and organise ideas and work practices, to find and engage others inside and outside the organisation [EG07].

The purpose behind using Microblogs is rather different from that of Blogs: very often, a Microblog is employed for a variety of work-related purposes even when it is not the optimal tool available [ZQCW10]. The major benefits reported include possessing awareness about what others are working on, making new connections [ZQCW10] and improving communication and organisational transparency [GKRS09]. Microblogging creates informal, social, group-structural and workspace awareness [BR09]. Microblog use on a corporate intranet differs from that of Twitter use on Internet: employees make use of organisational Microblogs (e.g. Yammer) more for publishing news about their groups or business units, rather than disseminating news pertinent to themselves [ZQCW10]. Microblogs are perceived as a skillful mechanism for generating virtual water-cooler conversations and are increasingly recognised as a new informal communication channel that complements other forms of interaction. [ZQCW10].

Arinia [BBKS10] and Communote [BR09] are two tools developed as a fast and secure internal alternative to email. However, the functionalities of these two tools are surprisingly similar to the utilitarian purposes of Microblogs. It is reported that these tools have been utilised as central information and communication channels within a company: they are an efficient alternative to email for internal communication [BR09]. It has been established that these tools do not altogether eliminate informal small talk during coffee breaks; on the contrary, they serve to enrich these talks, by providing an initial topic for discussion [BR09].

5.2.2. Challenges

The main challenges reported for the use of Blogs in organisations are related to motivation and adoption. Employees express frustration if they invest time in writing Blog posts and receive little or no perceived return on investment [YGB09]. It appears necessary to motivate employees, by making it very clear and apparent to them, that contributing to the company Blog is an opportunity and there is a social value attached to it, which can in turn lead to achieving a motivated and productive corporate culture [YGB09]. Promotion constitutes an important precondition for weblog success [ST08]. Managerial influences and social influences are also perceived to have a significant impact on the adoption of blog use [WRM09]. The usage of blogs by other employees in the same branch office is associated with higher blog usage by an individual employee [WRM09]. Adoption is negatively correlated with age, while there is no impact of gender on blog use [WRM09].

Prevalent corporate culture and the communication climate in an enterprise seem to determine whether or not Blog can be incorporated into the organisation, as it is very vital for both individual and corporate users to accept its characteristics of informal nature and openness [KHE07]. It therefore becomes essential to find a balance between formal guidance and employee self efficacy [KHE07]

Success of Microblogging is subject to participation and critical mass as well [ZQCW10]. A few of the perceived disadvantages of Microblogging include the issue of data protection, the loss of human interaction [BBKS10] and the very significant matter of privacy issues [GKRS09].

5.3. Wiki

A Wiki is described as a website that allows users to add and update content using their own Web browser. All studies about Wikis identified in this SMS are referenced in Table 5. Some papers analyse general purpose Wikis, whereas others investigate the usage of custom implemented ones, which have been developed for enhancing specific features or for overcoming some limitations [AGJP09] of standard Wikis.

5.3.1. Benefits

Wiki is a lightweight and all purpose tool, which can serve as a knowledge repository, a means for staging a project, a coordination mechanism, and a shared workspace [Kro08] and which can be readily set up and used effectively [HP07]. It is mainly used for knowledge management, especially for knowledge construction and information and document sharing [Ras09, Ram06], but in addition, it can be used as a coordination backbone, alternative to traditional Knowledge Management Systems [Hes10] and to raise awareness of global projects [GKP^{+}09].

Content is present in just one location, so it is easier to find, but locating specific information in a Wiki has been reported as an issue [RHH]. It is for this reason that Riki [RHH] has

Table 5: Papers on Wiki

Author	Title	Year
Arazy [AGJP09]	Wiki deployment in corporate settings	2009
Ben-Chaim [BCFR09]	An effective method for keeping design artifacts up-to-date	2009
Black and Jacobs [BJ10]	Using Web 2.0 to improve software quality	2010
Brzozowski et al. [BSH09]	Effects of feedback and peer pressure on contributions to enterprise social media	2009
Costa [CSN+09]	Social Knowledge Management in Practice: A Case Study	2009
Danis [DS08]	A wiki instance in the enterprise: opportunities, concerns and reality	2008
Gotel [GKSN08]	Working Across Borders: Overcoming Culturally-Based Technology Challenges in Student Global Software Development	2008
Gotel [GKP+09]	Evolving an infrastructure for student global software development projects: lessons for industry	2009
Happel [Hap09]	Social search and need-driven knowledge sharing in Wikis with Woogle	2009
Hasan and Pfaff [HP07]	Emergent Conversational Technologies that are Democratising Information Systems in Organisations: the case of the corporate Wiki	2007
Hasan et al. [HMPW07]	Beyond Ubiquity: Co-creating Corporate Knowledge with a Wiki	2007
Hester [Hes10]	Increasing collaborative knowledge management in your organization: characteristics of wiki technology and wiki users	2010
Holtzblatt et al. [HDW10]	Factors impeding Wiki use in the enterprise: a case study	2010
Knauss [KBKF09]	SmartWiki: Support for high-quality requirements engineering in a collaborative setting	2009
Krogstie [Kro08]	The wiki as an integrative tool in project work	2008
Krogstie [Kro09]	Using Project Wiki History to Reflect on the Project Process	2009
Majchrzak et al. [MWY06]	Corporate wiki users: results of a survey	2006
Phuwanartnurak [Phu09]	Did you put it on the wiki?: information sharing through wikis in interdisciplinary design collaboration	2009
Privman [PH08]	Whose (Partially Distributed) Team Are You On? Interviews About ”Us vs. Them” in Corporate Settings	2008
Radziwill [Rad04]	TWiki as a Platform for Collaborative Software Development Management’	2004
Raman [Ram06]	Wiki technology as a” free” collaborative tool within an organizational setting	2006
Ras [Ras09]	Investigating Wikis for software engineering - Results of two case studies	2009
Rech et al. [RHH]	Using Wikis to Tackle Reuse in Software Projects	2007
Tansey [TS10]	Annoki: a MediaWiki-based collaboration platform	2010
Turner et al. [TQB+10]	Exploring the workplace communication ecology	2010
Wu et al. [WYB10]	Finding Success in Rapid Collaborative Requirements Negotiation Using Wiki and Shaper	2010

been developed to add metadata for indexing the content and easily search for it. The history feature has been reported as one of the helpful characteristics of Wikis [KBKF09, Kro09], that supports recall and reflection; creates a knowledge repository; and assists in avoiding information overload.

In Software Engineering, Wikis possess a variety of uses and advantages. Some of these may be enumerated as requirement negotiation [WYB10], organising and managing documents, storing all project information in just one place and reusing their content, increasing the quality and the completeness of documentation and lowering the barriers for sharing knowledge [RHH, KBKF09]. Radziwill [Rad04] reports the usage of a Wiki as a platform for collaborative software development management, using it successfully for task assignments, status reporting, elucidating requirements and exploring designs. Since simultaneous work is possible, Wikis are also used for writing documentation collaboratively [KBKF09].

5.3.2. Challenges

Some of the issues reported for Wiki usage are related with the asynchronous nature of a Wiki, that requires the support of other channels to ensure the sharing of the content [Phu09]; with the lack of motivation for potential participants [AGJP09]; with the necessity to provide guidelines and policy of usage [HDW10]; and with the fact that they are appropriate mainly for knowledge that is ad hoc and dynamic [Hes10]. One common concern as well is the fact that there is no totally obvious picture about the specific corporate settings in which Wikis in reality thrive [AGJP09]. For instance, Software Engineering literature is not yet well defined and distinct about the phase of the Software Engineering process when Wikis actually become essential [Ras09]. To investigate this issue, WikiWinWin

[WYB10] and SmartWiki [KBKF09] have been implemented and successfully used for requirement negotiation and specification in Software Engineering, but further research is undeniably required to formalise how, in which situation and in which organisational context it is recommended to use Wikis.

A few of the other difficulties associated with Wikis are that often people are reluctant to modify content put up by others [DS08] or to share specific information due to a perceived extra cost; the nature of the information may also be a deterrent factor; the desire to share only finished content and finally, sensitivity to the openness of the sharing environment [HDW10]. Despite this fact, corporate Wikis democratise organisational information and knowledge [HP07, HMPW07], empowering employees with autonomy and freedom and allowing different users to voice their opinion and views, in a democratic manner [AGJP09]. Vandalism usually does not occur [DS08]. Effective use of the Wiki often depends on the organisational culture and the power structure [HP07] and how these are inclined to influence the ownership and authority. Enjoyment has been reported as the main driver for corporate Wiki participation, but it has been observed that in some cases strategies and incentives that encourage participation are also necessary [AGJP09]. Besides, it appears that further research is desirable on the motivation side [AGJP09].

Wiki is a very powerful tool, but often it is not sufficient in and by itself. It appears necessary to customise the available features [Ram06] or to develop ad hoc Wikis or plug-ins to overcome some of the issues identified in the literature. For example, the need for structuring the content and finding the information has been investigated by Ben-Chaim et al. [BCFR09], who report about a Wiki enhanced with a hierarchal glossary to maintain design artifacts, and by Happel, [Hap09], who proposes the introduction of social search and need-driven knowledge sharing. The necessity to improve the collaborative support feature of Wiki has been investigated by Tansey and Stroulia [TS10] with Annoki. This tool is intended to support collaboration by improving the organisation of content, managing access to content, assisting in the creation of content, and lastly, graphically displaying information about content stored on the Wiki. In addition, ResearchWiki [DS08] provides a basis for broader collaboration, enabling greater transparency in the work of a large distributed research organisation.

Practitioners tend to develop ad hoc implementations to add structure to Wikis and to improve their usability, since Wikis can be unintuitive and to make use of them advantageously, requires training [Ram06, WYB10]. Furthermore, it is often necessary to provide policies and guidelines about Wiki usage [HDW10] and to explain good practices. Some of these guidelines would be along the lines of explaining the nature of the content that should be included on the Wiki[PH08], and furthermore, encouraging employees to agree on how to use this tool within the team or even within the organisation.

5.4. Social Networking

A Social Networking Service (SNS) is an online website that focuses on facilitating the building and reflecting of social networks or social relations among people, who may share common interests and/or activities. A SNS consists of a representation of each user (often a profile), his/her social links, and a variety of additional services. In this review 14 studies were identified, which analysed the usage of SNS (see Table 6). The papers report mainly about the usage of internally developed Social Network Sites, like Beehive [SDEL09, WDM10, FDM+08, GDD+08, DGM+09, DMG+08] , WaterCooler [Brz09], ICT [WEK06], WBSN [CSN+09] and a photo sharing system [TSM09]. It is evident that most of the work has been conducted by IBM, since Beehive, ICT and the photo sharing system have been developed within IBM and the papers report about the usage of these SNS within IBM itself.

Further reports on the employment of corporate social networking and the experiences that are a result of this use, have been reported by Richter and Riemer [RR09], who conducted a study in three large multinational, knowledge intensive organisations, along with Brzozowski et al. [BSH09], who reported about the usage of SNS at Hewlett-Packard (HP). Moreover, a study about an undergraduate software engineering project using a plug-in for Facebook called CommonGround [CDMD10] was identified: the plug-in encourages interaction and status awareness among users. However, only a single case study with reference to workplace use of Facebook and LinkedIn was found [SG09].

5.4.1. Benefits

Organisational Social Networking Sites are treated differently from public ones, such as Facebook [DGM+09]. They have different patterns of use and different motivations governing their uses [DMG+08]. The major distinction in the pattern in which people use public SNS and organisational SNS is that, in a corporate context, users are inclined to attach greater importance to people search (establishing new relationships) and context awareness (getting to know someone better), whilst being less interested in exchange and maintaining network awareness [RR09]. However, in organisational settings, Facebook is used extensively to maintain awareness about colleagues' activities and to build rapport and stronger working relationships [SG09], while LinkedIn is widely used to build and maintain external professional networks [SG09]. Organisational Social Networking sites are not only used for socialising [SDEL09], but also for project planning, creating team awareness, and for fostering community building and group interactions. SNS are extensively utilised not only to build and foster connections among employees, thus reflecting real life relationships but also for the purpose of forging new connections with people not known in real life but belonging to the organisation [GDD+08]. This is an innovative method by which productive connections are formed in the company [SDEL09] a] and it also helps to identify experts in the organisation and how to get in contact and connect with them, in order to gain professional information and knowledge [RR09]. SNS are used to extend face-to-face communication among employees. From the company perspective, SNS can offer an overview of the relationships existing among employees and they are useful in revealing the true social networking structure of the company. Information of this nature facilitates management reorganisation, project funding

Table 6: Papers on Social Networking

Author	Title	Year
Brzozowski [Brz09]	WaterCooler: exploring an organization through enterprise social media	2009
Brzozowski et al. [BSH09]	Effects of feedback and peer pressure on contributions to enterprise social media	2009
Charlton [CDMD10]	Encouraging interaction and status awareness in undergraduate software engineering projects: The role of social networking services	2010
Costa [CSN+09]	Social Knowledge Management in Practice: A Case Study	2009
Dimicco [DMG+08]	Motivations for social networking at work	2008
DiMicco [DGM+09]	People Sensemaking and Relationship Building on an Enterprise Social Network Site	2009
Farzan [FDM+08]	Results from Deploying a Participation Incentive Mechanism within the Enterprise	2008
Geyer [GDD+08]	Use and Reuse of Shared Lists as a Social Content Type	2008
Richter [RR09]	Corporate Social Networking SitesModes of Use and Appropriation through Co-Evolution	2009
Skeels [SG09]	When social networks cross boundaries: a case study of workplace use of facebook and linkedin	2009
Steinfield [SDEL09]	Bowling online: social networking and social capital within the organization	2009
Thom-Santelli [TSM09]	Learning by seeing: photo viewing in the workplace	2009
Weisz [WEK06]	Synchronous broadcast messaging: the use of ICT	2006
Wu [WDM10]	Detecting professional versus personal closeness using an enterprise social network site	2010

allocation and other management-level decisions [WDM10]. In organisational SNS, employees typically like to share both personal and professional information [GDD+08, DGM+09]. One of the main themes reported by the papers is user motivation. It is possible to motivate employees to participate with point incentive mechanisms that aim to give a good reputation to users [FDM+08]. However, it is found that this is often not necessary, because the SNS grants the employee the freedom to build his own personal context, thereby gaining a role in the organisation, which therefore can impact on future career advancement and on the ability to convince co-workers to support ideas and projects. [DMG+08]. It has also been reported that feedback about content being seen by co-workers should be provided to content authors, which should motivate users to further contribute to enterprise social networks [BSH09].

Table 7: Papers on Social Bookmarking

Author	Title	Year
Black and Jacobs [BJ10]	Using Web 2.0 to improve software quality	2010
Damianos et al. [DCG+07]	Exploring the Adoption, Utility, and Social Influences of Social Bookmarking in a Corporate Environment	2007
Farrell et al. [FLN+07]	Socially Augmenting Employee Profiles with People-Tagging	2007
Millen et al. [MFK06]	Dogear: Social bookmarking in the enterprise	2006
Pan and Millen [PM08]	Information Sharing and Patterns of Social Interaction in an Enterprise Social Bookmarking Service	2008
Storey et al. [SRB+08]	TODO or to bug: exploring how task annotations play a role in the work practices of software developers	2008
Storey et al. [SRS+09]	How Software Developers Use Tagging to Support Reminding and Refinding	2009
Treude and Storey [TS09]	How tagging helps bridge the gap between social and technical aspects in software development	2009

5.4.2. Challenges

Some of the more important concerns reported in literature are associated with identity management and with multiple profile management [DGM+09]. Furthermore, contribution to productivity is often difficult to prove [SG09, Brz09], as issues of crossing hierarchy power boundaries are likely to arise, in addition to the risk of mixing personal and professional identities [SG09].

The only examples reported from a SE perspective are WSBN, used as an alternative tool for knowledge management [CSN+09], and CommonGround, which offers a means to foster group interaction and community building, to maintain interactive cohesiveness, team awareness and project planning beyond face to face meetings [CDMD10].

5.5. Social Bookmarking

Social Bookmarking is a method for Internet users to organise, store, manage and search for bookmarks of resources online. Tagging is a significant feature of social bookmarking systems, enabling users to organise their bookmarks in flexible ways and develop shared vocabularies known as folksonomies.

In the collection of papers for this study, seven articles have been selected that explore the use of Social Bookmarking in organisations. Among these works, three main categories can be identified: People Tagging system [FLN$^+$07], Source-code Tagging systems [SRB$^+$08, SRS$^+$09, TS09] and traditional Social Bookmarking systems that are used for tagging only Intranet resources [MFK06] or both Intranet and external Internet bookmarks [DCG$^+$07, PM08].

A range of unique advantages are reported for employees and companies utilising Social Bookmarking such as knowledge distribution [DCG$^+$07], awareness of the activities of employees in the organisation [FLN$^+$07] and expert finding [DCG$^+$07, FLN$^+$07], supporting informal processes of software development [TS09], resource management, information sharing and discovery and finally, social networking [DCG$^+$07]. As an information resource, social bookmarking systems have many potential integration points with other corporate applications [MFK06].

Many articles have highlighted the social nature of the tagging activity [SRB$^+$08, TS09, MFK06, FLN$^+$07, DCG$^+$07, PM08]. The number of public bookmarks has been found to be bigger than private bookmarks [PM08], thereby stressing upon the fact that people do consider tagging a social and collaborative activity.

All papers have a propensity to highlight the flexible, lightweight and bottom-up nature of tagging that allows users to create custom tag vocabularies. These characteristics are generally observed as one of the advantages of using tags for the function of categorising resources (independently if they are proper bookmarks, people or source codes). However, it has been reported that annotation can have different meanings, which are dependent on individual, team and community use [SRB$^+$08], so standardisation of tags can be introduced [MFK06] to obtain a shared vocabulary that permits information to be shared more effortlessly among members (for e.g. subscribing to a RSS feed related to a specific tag).

All papers retrieved analyse the usage of tools developed specifically for the enterprise: TagSEA for source code annotation [SRB$^+$08, SRS$^+$09, TS09], Fringe for people-tagging [FLN$^+$07], Dogear for internet bookmarks [MFK06, PM08] and Onomi for corporate intranet tagging [DCG$^+$07]. A particularly interesting point can be made that, from this review, there is no research that investigates whether and how people essentially use existing publicly available online Social Bookmarking systems in their working activities. This leads to an issue that is reported in some papers for e.g.[DCG$^+$07, PM08] about providing the right motivation to employees to use such systems. It emphasises the fact that to foster easier adoption, the repository should be initially populated.

5.6. Multiple Social Software

Most empirical studies identified in literature focus on the usage of one specific type of SoSo, without taking into consideration its relation with other SoSo and other communication channels available in the company. Only 9 papers (see Table 1) investigate more than one SoSo in the same study. Five papers analyse the use of more than one kind of SoSo but fail to provide findings about SoSo in general. Four papers do not consider features of each SoSo in detail, but provide an overview of the use of SoSo in general and how different SoSo relate to each other. These four papers are reported in Table 8 and they are discussed in detail in the next paragraph.

Table 8: Papers considering more Social Software

Author	Title	Year
Black and Jacobs [BJ10]	Using Web 2.0 to improve software quality	2010
Brzozowski [Brz09]	WaterCooler: exploring an organization through enterprise social media	2009
Brzozowski et al. [BSH09]	Effects of feedback and peer pressure on contributions to enterprise social media	2009
Turner et al. [TQB$^+$10]	Exploring the workplace communication ecology	2010

Turner et al. [TQB$^+$10] report about the concept of “communication ecology” and highlight the role of different SoSo for complementing the other communication channels available in the company. This work seeks to emphasise the importance of investigating the interrelations between tools, rather than focussing on one single tool at a particular time. This idea is further discussed in Subsection 6.3. With his study, Brzozowski [Brz09] affirms that SoSo presents an opportunity for an organisation to build a distributed knowledge base and thereby, increase the sense of connection employees experience towards the company’s initiatives and to each other. Black and Jacobs [BJ10] describe the added benefits of using SoSo in SE, reporting, among other beneficial characteristics, that SoSo *increase the quality of communication, increase sharing of knowledge and speed up the development.*

As SoSo provide unstructured channels for communication, SoSo can, and is being increasingly used for diverse purposes, besides being beneficial for communication, collaboration and knowledge management among team members. The specific usage appears to depend on local protocols.

6. Discussion

This section discusses the results of this Systematic Mapping Study (SMS): an integrated and synthesised answer is provided in response to the research questions; contributions to the Global Software Development (GSD) field are highlighted and as a final point, emerging themes that become visible when comparing different studies are discussed. The final part of the section involves a brief debate and deliberation on the limitations of the study.

6.1. Answers to the Research Questions

This section discusses how the data extracted from the reviewed studies address our research questions. By answering the three research questions, the study aims to provide a synthesised overview of the literature on the use of Social Software (SoSo) in distributed teams and Software Engineering (SE).

RQ1. *What are the fora in which research on SoSo in distributed teams and software engineering has been published to date?*

Literature on SoSo in distributed teams and SE is spread among different research communities and countless publications. The distribution of papers in publication fora is reported in Figure 4. Most of the papers appear in Computer-Human Interaction (CHI), Computer Supported Cooperative Work (CSCW), and Hawaii International Conference on System Sciences (HICSS) conferences; it is interesting to note that more than one third of the papers (37 studies) all appear in different venues.

The spread of the research over many different fora definitely magnifies and supports the need for a SMS of this kind, because in such a context, a manual search can become highly demanding and challenging. Moreover, the presence of studies relevant for GSD in so many different fora confirms that GSD is a multidisciplinary subject; therefore, experiences reported in related fields should be taken into account while conducting research in GSD.

RQ2. *How is SoSo reported to be used in distributed teams and in SE?*

For every kind of SoSo, a wide set of usages are usually examined. Figure 8 reports the usage analysed in the papers for each kind of SoSo, keeping in mind the fact that more than one kind of usage can also be reported in the same study. Contrary to the apparent high importance of the social aspect offered by SoSo, socialisation is not the most important usage reported. SoSo is mainly studied as a support for collaborative work, simultaneously fostering awareness, knowledge management and coordination among team members. The earliest studies published on SoSo as well as the maximum numbers of studies on a particular category of SoSo are on Instant Messaging (IM). Information relevant to all the different categories of SoSo has been published in different communities as well as in SE. However, a noteworthy point is that a significant volume of SE research focuses on the usage of IM and Wikis.

More than half of the papers investigate the communication aspect of SoSo, since all kinds of SoSo are basically considered communication tools. However, as SoSo is a term applied to widely different applications with different affordances - which basically comprise user determined communication and collaboration, and community building in common - this SMS also purports to illustrate that different kinds of SoSo give evidence for different user profiles. IM is widely used to provide awareness, and to collaborate and coordinate among distributed team members. Wiki, Blog and Social Networking Sites are mainly used for Knowledge Management. However, most of the papers do mention more than one usage of SoSo. Despite various differences, common themes can be identified when looking at research results, which will be discussed in greater detail in Subsection 6.3.

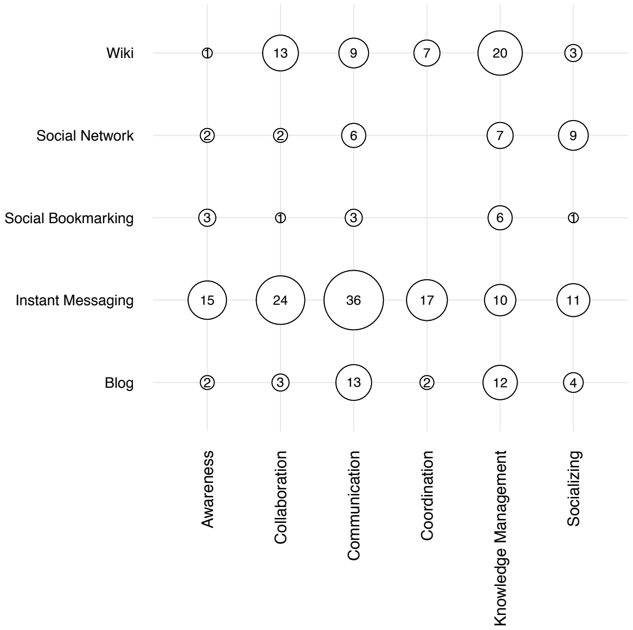


Figure 8: Usages for each SoSo

RQ3. *What are the benefits and the challenges of using SoSo in distributed teams and in SE?*

SoSo is used for different purposes, being beneficial for communication, collaboration and knowledge management among team members. In particular, SoSo is mainly used to exchange work-related messages, for articulation work, to coordinate projects and meetings and to negotiate colleagues' availability for discussions. Moreover, SoSo may perform the role of sharing organisational knowledge in an informal manner - Wikis and Blogs are frequently used as knowledge repositories and shared workspaces. Despite the fact that the primary use of workplace SoSo is for work discussions and not for social purposes, SoSo do go a long way in fostering social relationship among employees. SoSo can lead to spontaneous, informal conversations between team members and even go so far as to facilitate and maintain awareness among colleagues and to build stronger working relationships.

Additionally, it has been observed that in SE, SoSo is well-known for improving the quality of communication, increasing sharing of knowledge and speeding up of development. Furthermore, SoSo enables distributed developers to maintain both a general awareness of their entire team and to gain detailed and comprehensive knowledge of people they plan to work with. Some of the challenges encountered with regard to the usage of SoSo concern the need of a critical mass of users to motivate others to contribute to and use the tool. Willingness of employees to use SoSo for reporting to their superiors or to modify each other's content in Wiki is very often dependent upon the structure of the organisation. It appears mandatory to establish social conventions regarding the usage of SoSo in organisations or teams. Besides, data protection and privacy issues have been reported as other potential difficulties for the use of SoSo.

Table 9: Papers on GSD

Author	Title	Year	Fora
Arazy [AGJP09]	Wiki deployment in corporate settings	2009	Technology and Society Magazine
Ben-Chaim [BCFR09]	An effective method for keeping design artifacts up-to-date	2009	Wikis4SE Workshop
Erickson et al. [EKL+06]	A persistent chat space for work groups: the design, evaluation and deployment of loops	2006	Conference on Designing Interactive systems,
Gutwin et al. [GPS04]	Group awareness in distributed software development	2004	Conference on Computer supported cooperative work
Handel and Herbsleb [HH02]	What is chat doing in the workplace?	2002	Conference on Computer supported cooperative work
Knauss [KBKF09]	SmartWiki: Support for high-quality requirements engineering in a collaborative setting	2009	Wikis4SE Workshop
LaToza et al. [LVD06]	Maintaining mental models: a study of developer work habits	2006	International Conference on Software Engineering
Niinimaki and Lassenius [NL08]	Experiences of Instant Messaging in Global Software Development Projects: A Multiple Case Study	2008	International Conference on Global Software Engineering
Radziwill [Rad04]	TWiki as a Platform for Collaborative Software Development Management	2004	Proceedings of SPIE
Thissen et al. [TPBA07]	Communication tools for distributed software development teams	2007	SIGMIS Conference on Computer Personnel Research
Treude and Storey [TS09]	How tagging helps bridge the gap between social and technical aspects in software development	2009	International Conference on Software Engineering
Woerner et al. [WYO07]	Conversational Coherence in Instant Messaging and Getting Work Done	2007	Hawaii International Conference on Information Systems
Wu et al. [WYB10]	Finding Success in Rapid Collaborative Requirements Negotiation Using Wiki and Shaper	2010	Hawaii International Conference on Information Systems

6.2. SoSo in GSD

Table 9 reports on the papers retrieved for this review that focus on distributed software development teams. Only 12 of the 100 articles report explicitly on the use of SoSo in GSD. As already highlighted, research is spread among very different fora and only two of these papers have been identified during the pilot study (see Subsection 3.1).

The initial research question about how SoSo is used in GSD teams, which formed the basis for the motivation behind this entire SMS, can now be answered by looking at what research has investigated to date. From this review, Instant Messaging (IM) is the category of SoSo that is most widely researched in relation to GSD: IM is used in GSD to get quick answers and immediate feedback [LVD06, TPBA07], to facilitate multi-tasking and communication with numerous people simultaneously, as a secondary channel in meetings [NL08], and to maintain both a general awareness of the entire team and more detailed knowledge of people they plan to work with [GPS04]. The popularity of IM is confirmed by the work of Black et al. [2], that run a survey investigating the use of SoSo in GSD: thank to their work, it is possible to highlight that IM is not only the most researched SoSo but also the most used among GSD developers.

From this mapping study, it emerges that also Wikis are widely researched in relation to GSD: they are typically used for organising and managing documents, storing all project information in just one place and reusing their content, increasing the quality and the completeness of documentation and lowering the barriers for sharing of knowledge [KBKF09].

This review has made it possible to provide results from related fields that can give an indication as to the means by which SoSo is capable of supporting virtual teams in GSD. Major difficulties in collaboration, coordination and communication [5] can be alleviated by the usage of SoSo: IM can replace planned or impromptu face-to-face meetings, that are not always entirely feasible in GSD settings; Wiki can be used for knowledge sharing; Microblogs can generate virtual water-cooler conversations and are used as an informal communication channel [ZQCW10]. Despite this review did not reveal a considerable amount of research on Microblogs, this SoSo has been reported as widely used by GSD developers [2]. Other kinds of SoSo, such as Social Networking Sites (SNS) can be used for socialising [SDEL09], project planning, creating team awareness, and for fostering community building and group interactions. They also help to identify experts in the organisation and to get in contact and connect with them, in order to gain professional information and knowledge [RR09, GDD+08]. Social Bookmarking can also be beneficial for collaboration in GSD, since it fosters knowledge distribution [DCG+07], provides awareness of people's expertise [FLN+07] and supports expert finding [DCG+07, FLN+07]. Moreover, Social Bookmarking supports informal processes in software development [TS09], resource management, information sharing and discovery and social networking [DCG+07].

However, a systematic mapping study is only capable of mapping research that has been published to date. There is a belief that the use of SoSo in real world global software development is more widely spread than what is visible in actual research publications. The real challenge lies in answering the question about the reason behind why SoSo is favoured for distributed collaboration. That question will be answered in Subsection 6.3 on emerging themes.

6.3. *Emerging themes and future research*

The analysis of the papers exposed common issues across different research fields and also different kinds of SoSo and different use contexts. These issues are reported in this Subsection as emerging themes, in view of the fact that they can constitute interesting future research directions.

Ecology of communication channels

Though few of the studies address more than one kind of SoSo, many of them remark upon the necessity of examining more than one communication channel and the interaction among these various channels, when they are used by a team [TS10, TPBA07, Phu09, TQB+10, HDW10, ZR09, QC05, CW05, PH08]. Turner et al. [TQB+10] propose using the concept of "communication ecology" and consider SoSo part of a set of tools, which complements other available communication channels. They also highlight the role of different SoSo in this ecology. Bradner et al. use the term "interaction ecology" when analysing the adoption of the IM tool in IBM [BKE02].

The research conducted by the authors of this SMS highlighted the importance of the "ecology of channels" - of which SoSo is an integral part - analysing the role of IM in a GSD team [7]. After conducting this SMS, the need for further research in this direction is confirmed. While analysing the use of SoSo many studies refer to other tools, the connection and the usage of different channels by teams is not detailed, neither the complementary role of different channels is researched. By providing unstructured channels, SoSo and especially IM appear to posses the potential to both coordinate the use of other channels as well as to provide a media to negotiate communication channel usage. It is in this field that researchers can focus to understand how SoSo and other tools can be used effectively.

Socialization

The term Social Software used for the tools discussed in this study indicates their support for team collaboration, sharing and socialisation. Though Pi et al. [PLCL08] report positive effects of IM in respect to the satisfaction of both formal and informal communication in the work context, socialisation or usage of SoSo for non-work related topics is seldom reported. This might depend on the specific research design and research question as Cho et al. [CTK05] and Pauleen and Yoong [PY01] indicate that SoSo can be used explicitly to facilitate and maintain relationships.

In most of the research, though, the social seems to exist as a subtext to work related interaction. To cite an example, Handel and Herbsleb state about IM usage that *the content was primarily focussed on work task, with a smattering of non-work topics and humor* [HH02]. Other works report similar findings [QC05, AH06] and the authors own research confirms this finding [7]. Due to its perception as channel for informal communication, SoSo seems to invite more social subtext than other media and consequently, supports social relations and team spirit. Though this study does not promote the design of social relations due to ethical considerations, a better understanding of the role of social relationships and their development as part of any successful collaboration, may improve and augment the future design of tools, so as not to hinder informal communication and social subtext.

Appropriation and Structuring

The often cited advantage of SoSo as providing an unstructured communication channel also appears to provide a challenge. Throughout the reviewed articles, the need to develop structures, rules, good practices or agreements on how to use SoSo in work contexts is mentioned (See e.g.[TS10, Tan07, Phu09, HDW10, Hap09, GKRS09, KHE07, PH08]).

SoSo appears to afford conscious appropriation by teams and organisations, perhaps because of its lack of structure. Social protocols [19] that establish the use of SoSo need to be negotiated and renegotiated within the project or organisation. This appropriation in turn results in tailored support for the specific project rather than requiring the team to adjust to predefined structures.

The co-construction of social protocols guiding the usage of tools and communication channels need to be better understood. It is here that a third line of future research is visible. This might in turn help to understand the appreciation of these tools by software engineering practitioners.

The need of more Observational Studies

Last but not the least, there appears to be an imperative requirement of the need for more observational studies. The quantitative analysis of research methods applied in the articles indicates that only 11 of the studies included in this review used observational methods as part of their research. The emerging topics discussed above, pertain to varied issues, which are difficult to grasp through mere analysis of activity traces like logs, interviews and surveys, or experiments. They entail a requirement to relate the use of SoSo to the context in which it is used: other communication channels, the affordances of the development approach and project model used, among others. Studies such as these will provide a cogent understanding of the interplay between technical affordances and the social protocols developed, when adopting the tools in a specific context. Research conducted along these lines would be important to support future design and appropriation of collaboration support.

6.4. *Limitations of the Study*

Since this SMS collected papers merely published until July 2010, this appears to be the main limitation of the work, since research conducted subsequent to that period is not included in the review. As reported in Section 4, the Automatic Search has been performed the 22nd July 2010, hence the initial set of 4457 primary studies included in this SMS is limited to the papers that have been already published at that date. The selection process and the data extraction have been pretty demanding, consequently the time spent for analysing the data and writing the present article have been considerable. The decision of conducting a Manual Search to expand the initial set of primary studies, and the choice of analysing in detail the final set of papers raised the time required to finalise the present paper.

However, due to the fact that a map of the area until 2010 is provided and, owing to the broadness of the topic highlighted, this study could prove worthwhile and meaningful for performing further focussed Systematic Literature Reviews (SLR), on each category of SoSo and giving a detailed description on the usefulness of each tool.

The papers included in this review have been subjected to an accurate selection process and involved two researchers cross-checking the completeness and integrity of the data and validating the suitability of each paper for inclusion. However, the findings of this review may have been affected by the systematic bias in describing SoSo and the context of its usage. While endeavouring to include a wide selection of studies, this review may have missed some papers that address the use of SoSo practices in distributed teams and SE. However, an additional manual search was conducted on references of initial primary studies to extend the review and overcome potential systematic omissions.

7. Conclusion

The Systematic Mapping Study (SMS) reported here aims to provide a detailed overview on the use of Social Software (SoSo) in Global Software Development (GSD). SoSo is reported as being chiefly used as a support for collaborative work, fostering awareness, knowledge management and coordination among team members. Contrary to the evident high importance of the social aspects offered by SoSo, socialization is not the most important usage reported. The research is distributed over different communities and possesses inherent challenges and advantages. The challenges refer to the need to relate results and research paradigms from divergent communities to the existing research. The advantage relates to the possibility to deploy theories and research methods from different research traditions to make sense of the findings.

Three emerging themes based on an analysis of the body of literature were identified and these were: the appropriation and development of usage structures; understanding how an ecology of communication channels and tools are used by teams; and finally, the role played by SoSo - either as a subtext or as an explicit goal - in global software engineering. Research addressing these themes will not only provide more information on the usage of SoSo but will also aid in making enhanced support available for GSD and distributed development.

To conclude, the analysis revealed a surprisingly low percentage of studies including observations of the usage of SoSo. All the emerging themes identified for the purpose of the study required supplementary observational studies. It would appear advisable and worthwhile to take a step back and concentrate additional efforts in the understanding of the distributed practices, before focussing on new tools and process support.

8. Acknowledgement

The authors wish to thank Dr. Muhammad Ali Babar for his precious review of the protocol of this mapping study. They would like also to thank Paolo Tell, a colleague and a friend, for his prolific discussions on the topic and for his constant and unstinting research support.

Appendix A. Data Extraction Form

a) Paper Description

- Paper ID
- Bibliographic reference: Title, Year, Authors
- Publisher: ACM, IEEE, SpringerLink, Elsevier, AISeL, others
- Type of article: Journal/Conference/Workshop, ShortName, Publication Type
- Community: CHI, CSCW, SE, IS, EDU, others

b) Social Software

- Kind of Social Software: Blog, Wiki, Social Bookmarking, Instant Messaging, Social Network Site
- SoSo >= 1 (more than one Social Software)
- Name of SoSo

c) Company Background

- Kind of Company: Sw Company, Enterprise, IT Company, OSS , University Lab, Student project
- Distribution: Co-Located, Distributed, Globally Distributed, Not specified

d) Study Findings

- Themes: Communication, Coordination, Collaboration, Knowledge Management , Awareness, Negotiation, Socializing
- Methods: Qualitative, Quantitative
- Techniques: Interview, Survey, Activity log, Content Analysis, Experiment, Observations, Report own experience, User testing
- Usage of Social Software: Benefits and challenges *(discursive text extracts)*

Appendix B. Included Studies

[AFH08] Daniel Avrahami, Susan Fussell, and Scott Hudson. IM waiting: timing and responsiveness in semi-synchronous communication. In *CSCW '08: Proceedings of the 2008 ACM conference on Computer supported cooperative work*, 2008.

[AGJP09] O Arazy, I Gellatly, Soobaek Jang, and R Patterson. Wiki deployment in corporate settings. *Technology and Society Magazine, IEEE DOI - 10.1109/MTS.2009.932804*, 28(2):57–64, 2009.

[AH06] Daniel Avrahami and Scott Hudson. Communication characteristics of instant messaging: effects and predictions of interpersonal relationships. In *CSCW '06: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, 2006.

[BBKS10] S J Barnes, M Bohringer, C Kurze, and J Stietzel. Towards an Understanding of Social Software: The Case of Arinia. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on DOI - 10.1109/HICSS.2010.406*, pages 1–9, 2010.

[BCFR09] Y Ben-Chaim, E Farchi, and O Raz. An effective method for keeping design artifacts up-to-date. In *Wikis for Software Engineering, 2009. WIKIS4SE '09. ICSE Workshop on DOI - 10.1109/WIKIS4SE.2009.5069991*, pages 1–6, 2009.

[BFHB05] Jeremy Birnholtz, Thomas Finholt, Daniel Horn, and Sung Bae. Grounding needs: achieving common ground via lightweight chat in large, distributed, ad-hoc groups. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, 2005.

[Bie08] Matthew Bietz. Effects of communication media on the interpretation of critical feedback. In *CSCW '08: Proceedings of the 2008 ACM conference on Computer supported cooperative work*, 2008.

[BJ10] Sue Black and Joanne Jacobs. Using Web 2.0 to improve software quality. In *Web2SE '10: Proceedings of the 1st Workshop on Web 2.0 for Software Engineering*, 2010.

[BKE02] Erin Bradner, W.A. Kellogg, and Thomas Erickson. The adoption and use of Babble: A field study of chat in the workplace. In *ECSCW99*, number September, pages 139–158. Springer, 2002.

[BM02] Erin Bradner and Gloria Mark. Why distance matters: effects on cooperation, persuasion and deception. In *CSCW '02: Proceedings of the 2002 ACM conference on Computer supported cooperative work*, 2002.

[BR09] Martin Böhringer and Alexander Richter. Adopting Enterprise 2.0: A Case Study on Microblogging. *Mensch & Computer*, pages 293–302, 2009.

[Brz09] Michael Brzozowski. WaterCooler: exploring an organization through enterprise social media. In *GROUP '09: Proceedings of the ACM 2009 international conference on Supporting group work*, 2009.

[BSH09] Michael Brzozowski, Thomas Sandholm, and Tad Hogg. Effects of feedback and peer pressure on contributions to enterprise social media. In *GROUP '09: Proceedings of the ACM 2009 international conference on Supporting group work*, 2009.

[CDMD10] Terence Charlton, Marie Devlin, Lindsay Marshall, and Sarah Drummond. Encouraging interaction and status awareness in undergraduate software engineering projects: The role of social networking services. *Education Engineering (EDUCON), 2010 IEEE DOI - 10.1109/EDUCON.2010.5492580*, pages 179–184, 2010.

[CMRC01] Joanie Connell, Gerald Mendelsohn, Richard Robins, and John Canny. Effects of communication medium on interpersonal perceptions. In *GROUP '01: Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work*, 2001.

[CSN+09] Ricardo Costa, Edeilson Silva, Mario Neto, Diego Delgado, Rafael Ribeiro, and Silvio Meira. Social Knowledge Management in Practice: A Case Study. *Groupware: Design, Implementation, and Use*, pages 94–109, 2009.

[CTK05] H.K. Cho, Matthias Trier, and Eunhee Kim. The use of instant messaging in working relationship development: A case study. *Journal of Computer-Mediated Communication*, 10(4):00–00, 2005.

[CW05] Ann Frances Cameron and Jane Webster. Unintended consequences of emerging communication technologies: Instant Messaging in the workplace. *Computers in Human Behavior*, 21(1):85–103, January 2005.

[DCG+07] Laurie Damianos, Donna Cuomo, John Griffith, David Hirst, and James Smallwood. Exploring the Adoption, Utility, and Social Influences of Social Bookmarking in a Corporate Environment. In *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, pages 86–86. Ieee, January 2007.

[DFL08] E Diamant, Susan Fussell, and Fen-Ly Lo. Where did we turn wrong?: unpacking the effect of culture and technology on attributions of team performance. In *CSCW '08: Proceedings of the 2008 ACM conference on Computer supported cooperative work*, 2008.

[DFL09] E Diamant, Susan Fussell, and Fen-Ly Lo. Collaborating across cultural and technological boundaries: team culture and information use in a map navigation task. In *IWIC '09: Proceeding of the 2009 international workshop on Intercultural collaboration*, 2009.

[DGM+09] J.M. DiMicco, Werner Geyer, D.R. Millen, Casey Dugan, and Beth Brownholtz. People sensemaking and relationship building on an enterprise social network site. In *hicss*, pages 1–10. IEEE Computer Society, 2009.

[DMG+08] Joan DiMicco, David R. Millen, Werner Geyer, Casey Dugan, Beth Brownholtz, and Michael Muller. Motivations for social networking at work. In *Proceedings of the ACM 2008 conference on Computer supported cooperative work - CSCW '08*, page 711, New York, New York, USA, 2008. ACM Press.

[DS08] C. Danis and David Singer. A wiki instance in the enterprise: opportunities, concerns and reality. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 495–504. ACM, 2008.

[EG07] Lilia Efimova and Jonathan Grudin. Crossing Boundaries: A Case Study of Employee Blogging. In *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, pages 86–86. Ieee, 2007.

[EKL+06] Thomas Erickson, Wendy Kellogg, Mark Laff, Jeremy Sussman, Tracee Wolf, Christine Halverson, and Denise Edwards. A persistent chat space for work groups: the design, evaluation and deployment of loops. In *DIS '06: Proceedings of the 6th conference on Designing Interactive systems*, 2006.

[ES03] Margaret Elliott and Walt Scacchi. Free software developers as an occupational community: resolving conflicts and fostering collaboration. In *GROUP '03: Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*, 2003.

[ESK+99] Thomas Erickson, D.N. Smith, W.A. Kellogg, Mark Laff, J.T. Richards, and Erin Bradner. Socially Translucent Systems: Social Proxies , Persistent Conversation , and the Design of Babble. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, number May, pages 72–79. ACM, 1999.

[FDM+08] Rosta Farzan, Joan M. DiMicco, David R. Millen, Casey Dugan, Werner Geyer, and Elizabeth a. Brownholtz. Results from deploying a participation incentive mechanism within the enterprise. In *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, page 563, New York, New York, USA, 2008. ACM Press.

[FKSS04] Susan Fussell, Sara Kiesler, Leslie Setlock, and Peter Scupelli. Effects of instant messaging on the management of multiple project trajectories. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, 2004.

[FLN+07] Stephen Farrell, Tessa Lau, Stefan Nusser, Eric Wilcox, and Michael Muller. Socially augmenting employee profiles with people-tagging. In *Proceedings of the 20th annual ACM symposium on User interface software and technology - UIST '07*, page 91, New York, New York, USA, 2007. ACM Press.

[GDD+08] Werner Geyer, Casey Dugan, Joan DiMicco, David R. Millen, Beth Brownholtz, and Michael Muller. Use and reuse of shared lists as a social content type. In *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, page 1545, New York, New York, USA, 2008. ACM Press.

[GKP+09] Olly Gotel, Vidya Kulkarni, Des Phal, Moniphal Say, Christelle Scharff, and Thanwadee Sunetnanta. Evolving an infrastructure for student global software development projects: lessons for industry. In *ISEC '09: Proceedings of the 2nd India software engineering conference*, 2009.

[GKRS09] Oliver Günther, Hanna Krasnova, Dirk Riehle, and Valentin Schoendienst. Modeling Microblogging Adoption in the Enterprise. In *AMCIS 2009 Proceedings*, 2009.

[GKSN08] Olly Gotel, Vidya Kulkarni, Christelle Scharff, and Longchrea Neak. Working Across Borders: Overcoming Culturally-Based

Technology Challenges in Student Global Software Development. In *2008 21st Conference on Software Engineering Education and Training*, pages 33–40. Ieee, April 2008.

[GMKF04] Darren Gergle, David Millen, Robert Kraut, and Susan Fussell. Persistence matters: making the most of chat in tightly-coupled work. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, 2004.

[GPS04] Carl Gutwin, Reagan Penner, and Kevin Schneider. Group awareness in distributed software development. In *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, 2004.

[HAB$^+$02] James Herbsleb, David Atkins, David Boyer, Mark Handel, and Thomas Finholt. Introducing instant messaging and chat in the workplace. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, 2002.

[Hap09] Hans-Jörg Happel. Social search and need-driven knowledge sharing in Wikis with Woogle. In *WikiSym '09: Proceedings of the 5th International Symposium on Wikis and Open Collaboration*, 2009.

[HDKC08] Yu-Ting Caisy Hung, Nguyen Thi Thao Duyen, Wei-Chang Kong, and Ai-Ling Chua. Reexamining Media Capacity Theories Using Workplace Instant Messaging. *IEEE Transactions on Professional Communication*, 51(4):352–368, December 2008.

[HDW10] Lester Holtzblatt, Laurie Damianos, and Daniel Weiss. Factors impeding Wiki use in the enterprise: a case study. In *CHI EA '10: Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, 2010.

[Hes10] Andrea Hester. Increasing collaborative knowledge management in your organization: characteristics of wiki technology and wiki users. In *SIGMIS-CPR '10: Proceedings of the 2010 Special Interest Group on Management Information System's 48th annual conference on Computer personnel research on Computer personnel research*, 2010.

[HH02] Mark Handel and James Herbsleb. What is chat doing in the workplace? In *CSCW '02: Proceedings of the 2002 ACM conference on Computer supported cooperative work*, 2002.

[HMPW07] H Hasan, J A Meloche, C C Pfaff, and D Willis. Beyond Ubiquity: Co-creating Corporate Knowledge with a Wiki. In *Mobile Ubiquitous Computing, Systems, Services and Technologies, 2007. UBICOMM '07. International Conference on DOI - 10.1109/UBICOMM.2007.36*, pages 35–40, 2007.

[HP07] H. Hasan and C.C. Pfaff. Emergent Conversational Technologies that are Democratising Information Systems in Organisations: the case of the corporate Wiki. *Information systems foundations: theory, representation and reality*, page 197, 2007.

[HRS04] Elaine Huang, Daniel Russell, and Alison Sue. IM here: public instant messaging on large, shared displays for workgroup interactions. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, 2004.

[IWW$^+$02] Ellen Isaacs, Alan Walendowski, Steve Whittaker, Diane Schiano, and Candace Kamm. The character, functions, and styles of instant messaging in the workplace. In *CSCW '02: Proceedings of the 2002 ACM conference on Computer supported cooperative work*, 2002.

[JSU03] Michal Jacovi, Vladimir Soroka, and Sigalit Ur. Why do we ReachOut?: functions of a semi-persistent peer support tool. In *SIGGROUP Bulletin*, volume 24, 2003.

[JYO07] Anne Jackson, JoAnne Yates, and Wanda Orlikowski. Corporate Blogging: Building community through persistent digital talk. In *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, pages 80–80. Ieee, January 2007.

[KBKF09] E Knauss, O Brill, I Kitzmann, and T Flohr. SmartWiki: Support for high-quality requirements engineering in a collaborative setting. In *Wikis for Software Engineering, 2009. WIKIS4SE '09. ICSE Workshop on DOI - 10.1109/WIKIS4SE.2009.5069994*, pages 25–35, 2009.

[KHE07] M. Kosonen, K. Henttonen, and H.K. Ellonen. Weblogs and internal communication in a corporate environment: a case from the ICT industry. *International Journal of Knowledge and Learning*, 3(4):437–449, 2007.

[Kro08] B.R. Krogstie. The wiki as an integrative tool in project work. *COOP, Carry-le-Rouet, Provence, France*, (7491), 2008.

[Kro09] B R Krogstie. Using Project Wiki History to Reflect on the Project Process. In *System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference on DOI - 10.1109/HICSS.2009.496*, pages 1–10, 2009.

[LSEP05] John Linebarger, Andrew Scholand, Mark Ehlen, and Michael Procopio. Benefits of synchronous collaboration support for an application-centered analysis team working on complex problems: a case study. In *GROUP '05: Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, 2005.

[LVD06] Thomas LaToza, Gina Venolia, and Robert DeLine. Maintaining mental models: a study of developer work habits. In *ICSE '06: Proceedings of the 28th international conference on Software engineering*, 2006.

[MFK06] David Millen, Jonathan Feinberg, and Bernard Kerr. Dogear: Social bookmarking in the enterprise. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, 2006.

[MWY06] Ann Majchrzak, Christian Wagner, and Dave Yates. Corporate wiki users: results of a survey. In *WikiSym '06: Proceedings of the 2006 international symposium on Wikis*, 2006.

[NFM$^+$03] H Natsu, J Favela, A L Moran, D Decouchant, and A M Martinez-Enriquez. Distributed pair programming on the Web. In *Computer Science, 2003. ENC 2003. Proceedings of the Fourth Mexican International Conference on DOI - 10.1109/ENC.2003.1232878*, pages 81–88, 2003.

[NL08] T Niinimaki and C Lassenius. Experiences of Instant Messaging in Global Software Development Projects: A Multiple Case Study. In *Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on DOI - 10.1109/ICGSE.2008.27*, pages 55–64, 2008.

[NWB00] Bonnie Nardi, Steve Whittaker, and Erin Bradner. Interaction and outeraction: instant messaging in action. In *CSCW '00: Proceedings of the 2000 ACM conference on Computer supported cooperative work*, 2000.

[OM03] Jacki O'Neill and David Martin. Text chat in action. In *GROUP '03: Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*, 2003.

[OZDL10] Carol Ou, Xuepan Zhong, Robert Davison, and Yi Liang. Can instant messaging empower teams at work? In *Research Challenges in Information Science (RCIS), 2010 Fourth International Conference on DOI - 10.1109/RCIS.2010.5507296*, pages 589–598, 2010.

[PH08] Robin Privman and S.R. Hiltz. Whose (Partially Distributed) Team Are You On? Interviews About" Us vs. Them" in Corporate Settings. *Americas Conference on Information Systems*, 2008.

[Phu09] Ammy Phuwanartnurak. Did you put it on the wiki?: information sharing through wikis in interdisciplinary design collaboration. In *SIGDOC '09: Proceedings of the 27th ACM international conference on Design of communication*, 2009.

[PLCL08] Shih-Ming Pi, Yi-Chih Liu, Tsang-Yao Chen, and Shih-Hua Li. The Influence of Instant Messaging Usage Behavior on Organizational Communication Satisfaction. In *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual DOI - 10.1109/HICSS.2008.445*, page 449, 2008.

[PM08] Ying Xin Pan and David R. Millen. Information Sharing and Patterns of Social Interaction in an Enterprise Social Bookmarking Service. In *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, pages 158–158. Ieee, January 2008.

[PRKT05] M Perttunen, J Riekki, K Koskinen, and M Tahti. Experiments on mobile context-aware instant messaging. In *Collaborative Technologies and Systems, 2005. Proceedings of the 2005 International Symposium on DOI - 10.1109/ISCST.2005.1553327*, pages 305–312, 2005.

[PY01] David J. Pauleen and Pak Yoong. Facilitating virtual team relationships via Internet and conventional communication channels. *Internet Research*, 11(3):190–202, 2001.

[QC05] A QuanHaase and Joseph Cothrel. Instant Messaging for Collaboration: A Case Study of a High-Tech Firm. *Journal of Computer-Mediated Communication*, 10:1–20, 2005.

[Rad04] Nicole M. Radziwill. TWiki as a platform for collaborative software development management. In *Proceedings of SPIE*, pages 609–617. Spie, 2004.

[RaDH06] J. Rennecker, a.R. Dennis, and S. Hansen. Reconstructing the Stage: The Use of Instant Messaging to Restructure Meeting Boundaries. *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, 00(C):27a–27a, 2006.

[Ram06] M. Raman. Wiki technology as a" free" collaborative tool within an organizational setting. *Information systems management*, 23(4):59–66, 2006.

[Ras09] E Ras. Investigating Wikis for software engineering - Results of two case studies. In *Wikis for Software Engineering, 2009. WIKIS4SE '09. ICSE Workshop on DOI - 10.1109/WIKIS4SE.2009.5069996*, pages 47–55, 2009.

[RHH] Jörg Rech, Christian Höcht, and Volker Haas. Using Wikis to Tackle Reuse in Software Projects. *Ieee Software*, pages 1–8.

[RR09] Alexander Richter and Kai Riemer. Corporate Social Networking SitesModes of Use and Appropriation through Co-Evolution. *20th Australasian Conference on Information Systems*, (Schooley 2005), 2009.

[SCB00] Marc Smith, J. J. Cadiz, and Byron Burkhalter. Conversation trees and threaded chats. *Proceedings of the 2000 ACM conference on Computer supported cooperative work - CSCW '00*, pages 97–105, 2000.

[SDEL09] Charles Steinfield, Joan DiMicco, Nicole Ellison, and Cliff Lampe. Bowling online: social networking and social capital within the organization. In *C&T '09: Proceedings of the fourth international conference on Communities and technologies*, 2009.

[SFN04] Leslie Setlock, Susan Fussell, and Christine Neuwirth. Taking it out of context: collaborating within and across cultures in face-to-face settings and via instant messaging, 2004.

[SG09] Meredith Skeels and Jonathan Grudin. When social networks cross boundaries: a case study of workplace use of facebook and linkedin. In *GROUP '09: Proceedings of the ACM 2009 international conference on Supporting group work*, 2009.

[SL02] Ylva Segerstad and Peter Ljungstrand. Instant messaging with WebWho. *International Journal of Human-Computer Studies*, 56(1):147–171, 2002.

[SMH06] Jeremiah Scholl, John McCarthy, and Rikard Harr. A comparison of chat and audio in media rich environments. In *CSCW '06: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, 2006.

[SRB+08] Margaret-Anne Storey, Jody Ryall, R Bull, Del Myers, and Janice Singer. TODO or to bug: exploring how task annotations play a role in the work practices of software developers. In *ICSE '08: Proceedings of the 30th international conference on Software engineering*, 2008.

[SRS+09] M.-A Storey, J Ryall, J Singer, D Myers, Li-Te Cheng, and M Muller. How Software Developers Use Tagging to Support Reminding and Refinding. *Software Engineering, IEEE Transactions on DOI - 10.1109/TSE.2009.15*, 35(4):470–483, 2009.

[ST08] Alexander Stocker and Klaus Tochtermann. Investigating Weblogs in Small and Medium Enterprises: An Exploratory Case Study. *Knowledge Management*, pages 95–107, 2008.

[Tan07] John Tang. Approaching and leave-taking: Negotiating contact in computer-mediated communication. *Transactions on Computer-Human Interaction (TOCHI*, 14(1), 2007.

[TPBA07] M Thissen, Jean Page, Madhavi Bharathi, and Toyia Austin. Communication tools for distributed software development teams. In *SIGMIS CPR '07: Proceedings of the 2007 ACM SIGMIS CPR conference on Computer personnel research: The global information technology workforce*, 2007.

[TQB+10] Thea Turner, Pernilla Qvarfordt, Jacob Biehl, Gene Golovchinsky, and Maribeth Back. Exploring the workplace communication ecology. In *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, 2010.

[TS09] Christoph Treude and Margaret-Anne Storey. How tagging helps bridge the gap between social and technical aspects in software development. In *ICSE '09: Proceedings of the 2009 IEEE 31st International Conference on Software Engineering*, 2009.

[TS10] Brendan Tansey and Eleni Stroulia. Annoki: a MediaWiki-based collaboration platform. In *Web2SE '10: Proceedings of the 1st Workshop on Web 2.0 for Software Engineering*, 2010.

[TSM09] Jennifer Thom-Santelli and David Millen. Learning by seeing: photo viewing in the workplace. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, 2009.

[WDM10] Anna Wu, Joan DiMicco, and David Millen. Detecting professional versus personal closeness using an enterprise social network site. In *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, 2010.

[WEK06] Justin Weisz, Thomas Erickson, and Wendy Kellogg. Synchronous broadcast messaging: the use of ICT. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, 2006.

[WFS09] Hao-Chuan Wang, Susan Fussell, and Leslie Setlock. Cultural difference and adaptation of communication styles in computer-mediated group brainstorming. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, 2009.

[WRM09] Sunil Wattal, Pradeep Racherla, and Munir Mandviwalla. Employee adoption of corporate blogs: A quantitative analysis. *hicss*, pages 1–10, 2009.

[WYB10] Di Wu, Da Yang, and Barry Boehm. Finding Success in Rapid Collaborative Requirements Negotiation Using Wiki and Shaper. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on DOI - 10.1109/HICSS.2010.210*, pages 1–10, 2010.

[WYO07] Stephanie L Woerner, JoAnne Yates, and Wanda J Orlikowski. Conversational Coherence in Instant Messaging and Getting Work Done. In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on DOI - 10.1109/HICSS.2007.152*, page 77, 2007.

[YGB09] Sarita Yardi, Scott Golder, and Michael Brzozowski. Blogging at work and the corporate attention economy. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, 2009.

[YML+00] R. Young, L.a. Macaulay, P.J. Layzell, S. Drummond, C. Boldyreff, R. Bedson, S. Lees, and O.P. Brereton. Student group working across universities: a case study in software engineering. *IEEE Transactions on Education*, 43(4):394–399, 2000.

[ZQCW10] Jun Zhang, Yan Qu, Jane Cody, and Yulingling Wu. A case study of micro-blogging in the enterprise: use, value, and related issues. In *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, 2010.

[ZR09] Dejin Zhao and Mary Rosson. How and why people Twitter: the role that micro-blogging plays in informal communication at work. In *GROUP '09: Proceedings of the ACM 2009 international conference on Supporting group work*, 2009.

References

[1] N. Ahmadi, M. Jazayeri, F. Lelli, and S. Nesic. A survey of social software engineering. In *Automated Software Engineering-Workshops, 2008. ASE Workshops 2008. 23rd IEEE/ACM International Conference on*, pages 1–12. Ieee, 2008.

[2] S. Black, R. Harrison, and M. Baldwin. A survey of social media use in software systems development. In *Proceedings of the 1st Workshop on Web 2.0 for Software Engineering*, pages 1–5. ACM, 2010.

[3] S.A. Bly, S.R. Harrison, and S. Irwin. Media spaces: bringing people together in a video, audio, and computing environment. *Communications of the ACM*, 36(1):28–46, 1993.

[4] Erran Carmel. *Global software teams: collaborating across borders and time zones*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.

[5] M. Cataldo, M. Bass, J.D. Herbsleb, and L. Bass. On coordination mechanisms in global software development. In *Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on*, pages 71–80. IEEE, 2007.

[6] E.Ó. Conchúir, P.J. Ågerfalk, H.H. Olsson, and B. Fitzgerald. Global software development: where are the benefits? *Communications of the ACM*, 52(8):127–131, 2009.

[7] Y. Dittrich and R. Giuffrida. Exploring the role of instant messaging in a global software development project. In *Global Software Engineering (ICGSE), 2011 6th IEEE International Conference on*, pages 103–112. IEEE, 2011.

[8] M.G. Farkas. *Social software in libraries: building collaboration, communication, and community online*. Information Today, Inc., 2007.

[9] R. Giuffrida and Y. Dittrich. Exploring the role of social software in global software development projects. In *Global Software Engineering Workshop (ICGSEW), 2011 Sixth IEEE International Conference on*, pages 108–110. IEEE, 2011.

[10] T. Greenhalgh and R. Peacock. Effectiveness and efficiency of search methods in systematic reviews of complex evidence: audit of primary sources. *BMJ*, 331(7524):1064–1065, 2005.

[11] J.D. Herbsleb and D. Moitra. Global software development. *Software, IEEE*, 18(2):16–20, 2001.

[12] E. Isaacs, A. Walendowski, and D. Ranganthan. Hubbub: A sound-enhanced mobile instant messenger that supports awareness and opportunistic interactions. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, pages 179–186. ACM, 2002.

[13] A.M. Kaplan and M. Haenlein. Users of the world, unite! the challenges and opportunities of social media. *Business horizons*, 53(1):59–68, 2010.

[14] Barbara Kitchenham and Stuart Charters. Guidelines for performing systematic literature reviews in software engineering. *Version*, 2(EBSE 2007-001):200701, 2007.

[15] M. Koch. Cscw and enterprise 2.0–towards an integrated perspective. *21th Bled eConference, eCollaboration: Overcoming Boundaries Through Multi-Channel Interaction*, 2008.

[16] I. Lykourentzou, F. Dagka, K. Papadaki, G. Lepouras, and C. Vassilakis. Wikis in enterprise settings: a survey. *Enterprise Information Systems*, 6(1):1–53, 2012.

[17] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson. Systematic mapping studies in software engineering. In *12th International Conference on Evaluation and Assessment in Software Engineering*, pages 71–80, 2008.

[18] P. Raeth and S. Smolnik. Antecedents and consequences of corporate weblog usage in the intranet: A process perspective. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pages 1–10. IEEE, 2010.

[19] Kjeld Schmidt and Carla Simonee. Coordination mechanisms: Towards a conceptual foundation of cscw systems design. *Computer Supported Cooperative Work (CSCW)*, 5:155–200, 1996. 10.1007/BF00133655.

[20] I. Steinmacher, A. Chaves, and M. Gerosa. Awareness support in global software development: A systematic review based on the 3c collaboration model. *Collaboration and Technology*, pages 185–201, 2010.

You Can Not Ask What You Do Not Suspect - An Argument for Observational Studies in Global Software Development Research

Rosalba Giuffrida

IT University of Copenhagen - Software and System Section
Copenhagen, Denmark rogi@itu.dk

Yvonne Dittrich

IT University of Copenhagen - Software and System Section
Copenhagen, Denmark ydi@itu.dk

Helen Sharp

Open University Walton Hall, Milton Keynes
MK7 6AA, UK Helen.Sharp@open.ac.uk

Abstract

Cultural, socio-technical and human aspects are an essential part of Global Software Development (GSD) activities. Investigating such aspects of everyday practices is challenging, as they are often not accessible solely through surveys and interviews. This paper maps the empirical research methods adopted to investigate GSD issues to date, highlights the low adoption of observational studies in GSD research and it argues for more, studies of this kind. The main challenge in performing observations in GSD is that teams are spatially distributed, thus observations need to take place not only in one single site as happens with co-located software teams, but in multiple globally-distributed sites along with the virtual space, where most of the collaboration occurs. This paper classifies the possible spatial distributions of GSD teams, discusses the challenges of performing observational studies in these different settings, and suggests how to overcome them.

1. Introduction

Global Software Development (GSD) is increasingly becoming a common practice in the software industry [11]. Organizations establish global software projects, which are scattered all around the globe, involving multiple teams located at different sites [27]. Since GSD is highly geographically dispersed, teams have to deal with temporal, geographical and socio-cultural distances [10]. Performing research in GSD is challenging, as researchers also have to deal with these distances, and there are few methodological discussions on how to overcome the associates problems - see [36] and [35] for exceptions.

In addition, Software Engineering (SE) is a cooperative work [16], so human aspects of SE need to be taken into consideration when investigating SE practices. This becomes particularly true when SE activities are distributed: cultural, socio-technical and human factors are an essential part of GSD practices. Most GSD research to date makes use of interviews and surveys as

Preprint submitted to Elsevier *March 12, 2014*

data collection techniques [46]. However, it is difficult to capture the subtleties of cultural, socio-technical and human factors through self-reports only, especially when implicit behaviour is to be captured, i.e. behaviour that is internalized to such an extent that the acting person himself is no longer aware of it. In his discussion about research methods, Robson [38] explains that "a natural and obvious technique to capture the actions and behaviour of people is to watch what they do, to record it and then to describe, analyse and interpret what we have observed". Following Robson's perspective, direct observations [38] allow to obtain an understanding of work practices, especially when social and human aspects are crucial, as happens in GSD. We use the term direct observations to span a wide range of methods that gather data contemporaneously, as work happens: situated and online observations, short term and long term observations, participant and non-participant observations. In short, any research method involving immediate observations of everyday software development activities. Direct observations are in contrast with self-reporting techniques in which practitioners tell about their experience either discursively through open-ended questions or interviews, or through quantitative surveys.

This paper provides an argument for the usage of more direct observations in GSD, in contrast with the low adoption of this kind of method in previous GSD research. A mapping study is performed to show which methods have been used to conduct empirical research in GSD to date, revealing an extensive usage of interviews and surveys and a limited adoption of observational studies[1]. This paper reports examples of observational studies in GSD, deriving both from the literature and from examples of the authors' research experiences, to show the importance of observational studies in GSD. In these examples, observational studies are used as the central data collection technique or with a supportive role for other techniques. However, the paper acknowledges that the spatial distribution of team members is the main challenge that researchers encounter when performing direct observations in distributed settings: in GSD projects, online environments play a crucial role and several different globally distributed physical field sites are involved, thus the concept of field site where observations should take place needs to be extended. A classification deriving from literature of possible spatial distributions of GSD teams is provided and suggestions on how to deal with the challenges caused by the spatial distribution are proposed.

2. Motivations

Two of the authors of the present paper performed a Systematic Mapping Study on the role of Social Software in GSD [20], which revealed a low adoption of observational studies. In the study, papers from different research traditions were mapped: in Figure 1 the distribution of methods in each research community is reported. In the Software Engineering (SE) community all methods are adopted, but observational methods are mainly - but not extensively - used in Human-computer Interaction (CHI) and Computer supported Cooperative Work (CSCW) communities. Interviews and surveys are widely adopted in all communities, but they are the main technique used in SE. The literature review [20] motivated us to further explore the topic and investigate what research methods are adopted to explore GSD issues. Smite et al. [46] conducted a Systematic Literature Review collecting empirical work in GSD published before November 2007; among other findings, the study highlights a wide usage of interviews and surveys, while only four papers, out of 59 studies included in the review, made use of observations. Thus, we

[1] We consider as observational studies all the studies that make some use of direct observations.

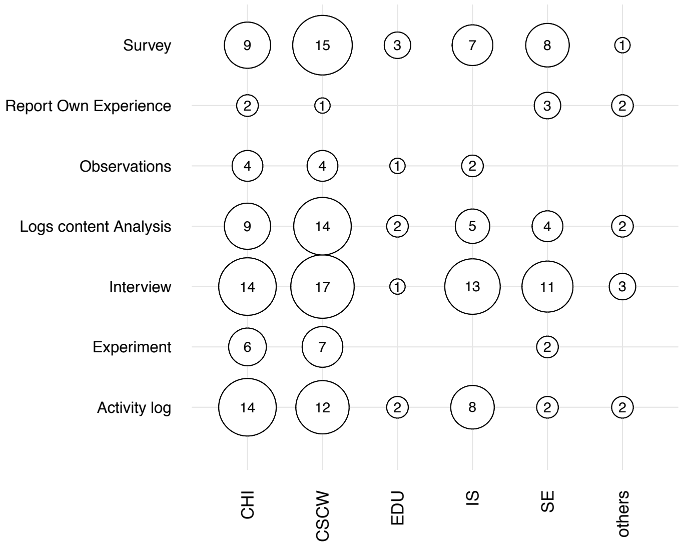


Figure 1: . Research Methods and Communities - data from Systematic Mapping Study on the use of Social Software in GSD [20].

decided to extend Smite et al.s study to check what are the research methods adopted to explore GSD issues nowadays. We collected research papers from 2007 until 2013 from the main venue for GSD research (International Conference on Global Software Engineering - ICGSE), and from the main venue for Empirical Research (Empirical Software Engineering Journal - ESE journal). In recent years, the situation did not change and observational studies are still not widely adopted, while GSD research is mainly carried on though surveys and interviews. This paper argues for the need of more observational studies due to their current scarce adoption, highlighting the advantages of direct observations in GSD, along with suggesting how to deal with challenges of observing GSD teams, physically distributed in multiple sites.

3. Related Work

3.1. Research Methods in Software Engineering

Empirical research methods are well-established as part of the Software Engineering research method toolkit. After developing and establishing quantitative empirical research in the late 80es and beginning of the 90es [2], the discussion of qualitative research methods started in 1999 with the article by Seaman [39] that suggests that qualitative methods can be used fro formulating hypothesis and continued in 2000 with the workshop “Beg, Borrow, or Steel: using multidisciplinary approaches in empirical software engineering research” [44]. Landmarks of this discussion are a special issues on Qualitative Research in Software Engineering [16] and the Workshop series on Cooperative and Human Aspects of Software Engineering (CHASE)[2], since

[2] `http://www.chaseresearch.org`

2008 at the International Conference on Software Engineering (ICSE)[3]. Qualitative empirical methods have been promoted to be able to understand software development as social practices. This article does not join the discussion on quantitative versus qualitative research, but argues for the usage of observational studies as opposed to experiments, surveys and interviews.

Experiments and tests place the observees in an artificial situation in order to measure the effect of specific treatments or to evaluate a method, technique or tool. The goal is the proof and/or quantification of causal relationships between the treatment of the free variable and the outcome of the dependent variable, while as much as possible all other influences are controlled [49]. Experiments can collect quantitative and qualitative data. An example for an experiment in relation to GSD is the work by Hess and Audy [23]. Due to the effort on the observees' side, experiments are often short term and - even when the experiments address group behavior - do not allow for the development of a practice [33], an established way of acting and interacting in the project. History and experience are either treated as biases or as part of the controlled variables.

Interviews and surveys ask respondents about facts or about their individual understanding of a situation. Both in their quantitative and qualitative form, they depend on the interviewer formulating questions about the area of concern. While quantitative surveys and interviews aim at statistical data and - if possible - statistical correlations between specific, predefined answers, qualitative surveys and answers allow for a co-construction of what is important with respect to the area of the interview [38]. The most prominent example for qualitative interviews is the article by Curtis et al. [12] that comes to the result that "the thin spread of application domain knowledge, fluctuating and conflicting requirements, and communication bottlenecks and breakdowns" (p. 1268) affect software quality and the software process. For GSD, the study by Estler et al. [19] provides a good example of usage of interviews analyzed both quantitatively and qualitatively. Interviews have the disadvantage that they are subject to the respondent bias. Interviewees might not be honest in their answers, especially if the questions address areas of ethical and/or legal consequences, or relate to the "dos and dont's" of the organization the research takes place in. Furthermore, the interviewee might not be able to account for what actually happened. Already Curtis et al. [12] argue for "the need for ecological data in technology research" (p. 1268), and regarded their interview study as only a starting point for such studies "on how characteristics of the situation affected human behavior" (p. 1285). Such studies require direct observations. As the literature studies [46] and [20] show, such observational studies are - more than 20 years after the article by Curtis et al. was published - still rare in SE and in GSD research.

Real world data can be document traces of collaboration, e.g. in form of electronic documents, chat logs, code structures, as well as good old in-situ participatory observation. Electronic traces of collaboration, though, need to be complemented with other methods. Aranda and Venolias article on the secret life of bugs [1] compares the amount of information about communication and collaboration around bug fixing gathered through different methods. The comparison provides a strong argument not to rely only on electronic traces; however, already human analysis of electronic traces is substantially better than algorithmic analysis.

3.2. What are Observations?

We use the term direct observations to span a wide range of methods: situated and online observations, short term and long term observations, participant and non-participant observations;

[3] `http://www.icse-conferences.org`

in short, any research method including observations of real-world software development. In-situ observations are typically documented by fieldnotes but can also be captured on audio or video - see the work conducted by Boden et al. [4] as an example in GSD. In-situ observations yield data about everyday life. Even if the research question requires interviews, surveys or statistical analysis for individual projects, or in order to compare projects, observational studies might be necessary to gain deeper insights or to triangulate findings. As our title states, it is difficult to ask about something you do not suspect. Observational methods allow understanding what in software projects actually happens, rather than what we researchers think happened.

However, observations are neither easy nor trouble-free: the major concern is that they require a heavy investment of time and effort [38]. Furthermore, in SE, observational studies are affected by the challenges of other qualitative studies, such as: gaining access to companies, dealing with trust acquisition, being allowed to collect data from documents [36]. Another issue of performing observations in SE concerns the extent to which an observer affects the situation under observation; this can be overcome by, for example, letting the observed person getting accustomed to the presence of the observer [38]. A practical challenge with observations is that becoming a participant observer [38] might be only possible if the researcher has the right technical background [17]. Furthermore, in SE, qualitative methods, as direct observations, are often considered softer or fuzzier than quantitative results [39]: this is a major concern that researchers have to deal with. Finally, observations can be seen as inappropriate for software development as it is considered an activity that takes place inside a persons head [39]; however, software developers reveal their thought processes when communicating with other developers [36], thus as a cooperative work [16] Software Engineering can be likewise observed. Challenges of observations in SE are well known and solutions about how they can be overcome have been suggested in several research papers - e.g. [17], [36], [38], [39]. The following subsection reports further discussions on the difficulties of performing qualitative studies in GSD.

3.3. Challenges of Qualitative Studies in GSD

Specific challenges of performing qualitative studies in GSD are not yet widely investigated. To the best of the authors' knowledge, there are solely two studies analyzing this topic: one recent article by Prikladnicki et al. [36] based on the authors past experiences, and a methodological reflection paper by Patil et al. [35], in which the authors reflect on the methods they adopted while studying a globally distributed project. In both studies, cultural and linguistic issues are considered, among others, the most important challenges that a researcher can encounter and that are distinctive for qualitative research in GSD projects. It can be necessary to adjust the research methodology to the cultural norms and customs of the remote site: there can be potential bias of the researcher, related to his own culture, and it may be difficult to follow local communication as team members may tend to use their native tongue when talking to each other [36]. Despite these challenges are reported for qualitative studies in GSD, to a larger extent they also affect observational studies. However, additional issues specific for performing observational studies in GSD are not explored yet in any research methodological reflection. These issues can be dealt with making use of recommendations offered by discussions carried on in social science [38] . This paper suggests that the primary challenge in performing direct observations in GSD is the geographical distribution of participants: the researcher can not be physically present in different sites at the same time. In the following, a brief summary of related works that can be beneficial for understanding how to perform observational studies in distributed settings is presented. We aim to learn from the discussions carried on in ethnographic research that involves a considerable amount of direct observations.

3.4. Observational Studies in Virtual World and in Distributed Settings

Performing observations in GSD is challenging due to the geographical distribution of the teams. The concept of field site [38] where observations take place needs to be extended: on the one hand, virtual is the space in which the team acts and here the observations should take place; on the other hand, the different globally distributed physical workplaces need to be taken into consideration. Software team members are immersed in a hybrid world [25] where the digital world, that supports the physical work of the team, needs to be investigated as a complementary field site for the (multi) physical one(s) - see e.g. Sharp et al. [42]

Concerning the former, it might be possible for the researchers to be virtually present with most or all participants in an online environment, observing team members as researchers do when adopting the virtual ethnography approach [24]. Since team members interact online, also the researcher can experience the same way of interaction. This is a common practice in online game research [5] as well as in Open Source Software (OSS) research [43], where researchers make use of observations of the virtual world, which is often the only space where the member meet regularly.

Discussing the latter, the primary challenge in performing observations in distributed settings is that the researcher can not be physically present in different sites at the same time. In mid-nineteens, ethnographers, anthropologists and sociologists faced similar challenges researching topics such as globalization, migrations, nationalism and other issues, not typically present just in one single site [28]. Ethnographic studies involve a considerable amount of direct observations, thus the solutions proposed by these researchers can inspire how to perform observational studies in distributed settings. Traditionally, observations of human behavior were carried on in a delimited geographical space by engaging in face-to-face interactions; Marcus [28] investigated how to adapt local ethnographic methods to global settings. The approach is called multi-site ethnography [28] and it is based on the idea that observational studies can involve more than one field site and the researcher can move through and between field sites. Later, Marcus extended his theories suggesting an approach called strategically-situated ethnography [29], that relies on the fact that participants themselves are also trying to manage scale and distribution from a single site: instead of trying to be everywhere, the researcher identifies the key sites or events where participants are working and from where they make their own activities intelligible for others; researchers can observe participants from the strategically-situated sites identified.

All approaches illustrated in the present section can inspire GSD researchers suggesting how to deal with challenges related to geographical distance when performing observational studies in GSD. This will be further discussed in Section 7 and ethnographic approaches for distributed settings will be used to suggest how to observe GSD teams.

4. Method

We conducted a Systematic Mapping Study (SMS) of empirical research papers in GSD to show which research methods are used to conduct empirical research in GSD. Papers published until 2007 are mapped in a study by Smite et al. [46], thus they are not included in the SMS we performed, that collects papers from 2008 until 2013. The results of Smite et al. [46] work are reported in Table 1, to complement the results of our SMS, that aims to extend their work including more recent papers. The design of the protocol for our SMS is described in the following.

The research question that motivates the present study is: “Which empirical methods are used to investigate GSD issues?” The SMS is performed on one conference and one journal:

namely International Conference on Global Software Engineering (ICGSE) and Empirical Software Engineering journal (ESE). The first venue is chosen as it is considered the main venue for GSD research, while the latter is the major venue for Empirical Software Engineering research, in which research about GSD is also published. Papers from 2007 and 2013 have been searched manually in the Proceedings of ICGSE and ESE. The search was performed in November 2013. The Digital Library of Association for Computing Machinery (ACM) was used to retrieve the papers. Only main conference papers are included; thus, workshop and position papers are not considered in the analysis. Primary studies are selected for inclusion reading the title and the abstract and checking whether the paper reports an empirical study about GSD. Both student projects and industrial papers are included. Papers without empirical validation in company or student settings or solely based on proof of concepts are excluded - e.g. literature reviews, theoretical papers, proposals of new methodologies, and presentation of frameworks. Papers that do not explicitly declare to investigate global or distributed projects are excluded as well.

Data required for the analysis are extracted by exploring the full-text of each included paper. In particular, the section dedicated to methodology is read - usually referred as: “data collection”, “method”, “methodology”. The SMS synthesizes data about whether the empirical study make use of qualitative, quantitative methods or both, and which data collection techniques are adopted: interview, survey, document analysis, experiment, observations, report own experience. The classification schema is taken from the one developed in a previous study by Giuffrida and Dittrich [20], in which data about research methods have also been analysed; the schema is compatible with the one adopted by Smite et al. A spread sheet is used for data extraction and data collection. The extraction of the data is based on what it is stated in the paper and it is not interpreted by the researcher that performed the review. When “case study” is reported as a research method, then the data collection method used during the case study is reported; the “case study” method is not further considered as a method on its own.

The purpose of the SMS is to provide a map of the empirical methods used to perform research in GSD. An additional qualitative analysis of papers that use observations is performed to provide an overview of the possible usage of observations for GSD research. The papers are classified according to the role of observations expressed in the methodological section. When the methodological section does not provide sufficient information, the full text is read and the analysis of the article is examined, in order to understand the actual usage of observations. When in doubt, the paper is classified as N/A. The classification schema is developed incrementally while reading the papers. The classification has been performed by one of the authors; another author checked a random sample of the papers classified; finally, all authors agreed on the classification schema.

5. Results of SMS

5.1. Research Methods in GSD

Table 1 reports the number of papers analyzed in this SMS that maps papers from ICGSE conference and ESE journal from 2008 until November 2013 and in a previous study by Smite et al. [46] that maps empirical papers in GSE until 2007. A total of 141 papers[4] have been included in this Systematic Mapping Study that classifies the research methods adopted in GSD research

[4] The Excel file with the references of included papers and related data extracted is available at the address: `www.itu.dk/people/rogi/icgseResearchMethods.xls`

Table 1: Number of Papers analysed by Smite et al. [46] and in this SMS.

Year	Number of Papers	Number of Papers Included
until 2007 Smite et al. [46]	n/a	59
ICGSE 2008	32	24
ICGSE 2009	29	23
ICGSE 2010	30	20
ICGSE 2011	26	23
ICGSE 2012	31	21
ICGSE 2013	27	25
ESE journal	n/a	5

Table 2: Distribution of Qualitative and Quantitative Studies.

Method	ICGSE	ESE	Smite et al. [46]	Total
Quantitative	37	3	n/a	40
Qualitative	90	3	n/a	93
Mix quantitative and qualitative	22	1	n/a	23

works. A synthesis of data extracted in this SMS and data about research methods presented by Smite et al. [46] are reported in Table 2 and Table 3. Qualitative methods have been widely used; however, quantitative and qualitative methods have been rarely used in combination. The techniques most widely adopted are interviews and surveys; a relevant portion of research consists of industrial papers, reporting practitioners' experience. Document analysis is often used in combination with other research methods, in particular with interviews and observations. Few studies make use of observations as an empirical method and, when they are adopted, observations are always used in combination with other research methods. The usage of observations is further described in the following paragraph.

Table 3: Empirical Research Methods adopted.

Method	ICGSE	ESE	Smite et al. [46]	Total
Interview	60	3	28	91
Survey	43	3	11	57
Document analysis	28	3	15	46
Report own experience	38	1	n/a	39
Experiment	13	1	8	22
Observations	15	1	4	20

5.2. The Role of Observations in GSD research

From the data reported in the previous paragraph, it is evident that interviews, surveys and experience reports are the most widely adopted research methods for investigating GSD field. On the contrary, the amount of observational studies is very low. Table 4 reports the 20 studies that make use of observations. Six out of 20 studies are published in 2013, showing, perhaps, a recent growing interest in this data collection technique. Observations are generally used in combination with other research methods, generally both interviews and document analysis, primarily for triangulating purposes. Thus, we classify two types of observational studies, depending whether observations are:

- *Central*, when observations are the main data collection technique, supported by other research methods that serve to triangulate findings (other methods such as document analysis and interviews are used for triangulate findings gained through observations);
- *Supportive*, when observations are used in preparation for other research methods and/or for complementing the findings of other research methods (observations are used for triangulate findings obtained by document analysis and interviews, that are the main data collection techniques).

Observations are in most of the cases used in the initial phase of the project in order to gain initial insights that will be further investigated, through additional observations and/or other research methods. Interviews generally follow observations, as observations provide an initial understanding for the project that suggests to researchers what to ask to practitioners as to say, researchers can not ask what they do not suspect.

The next section provides example of the two usages of observations identified (central and supportive), showing how this data collection technique can be beneficial for GSD research.

6. The Importance of Observational Studies: four Examples

In this section, we provide examples of observational studies taken from the authors research experience. The aim is to illustrate in detail how observational studies can be used to investigate GSD issues; in Examples 1 and 2 observations are used with a supportive role, while Example 3 makes use of observations as the main data collection technique. The authors reflect on their own research experience, arguing about the benefits of direct observations in GSD settings. The different cases are presented independently in the following and discussed together at the end of the section.

6.1. Example 1

Giuffrida and Dittrich [21] conducted an observational study about two novice GSD student teams collaborating between Denmark and China. One researcher travelled to each site and observed the Danish site for 4 months and the Chinese site for 10 days. Data was collected through document analysis, semi-structured interviews and direct observations of the weekly meetings among remote teams. Observations were used to support - confirming/disconfirming - findings gained through document analysis and for preparing interviews with team members. In one of the team, misunderstandings occurred during the weekly Skype meeting: Chinese students were often talking in Chinese, without informing the Danish team about the reasons for that. Danish students were puzzled by this behaviour and whenever they asked for clarifications, Chinese

Table 4: Observational studies in GSD research.

Authors	Year	Title
Damian and Zowghi [14]	2002	The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization
Taxen [48]	2006	An integration centric approach for the coordination of distributed software development projects
Damian et al. [13]	2007	Awareness in the wild: Why communication breakdowns occur
Korkala and Abrahamsson [26]	2007	Communication in Distributed Agile Development: A Case Study
Bugde et al. [7]	2008	Global Software Servicing: Observational Experiences at Microsoft
Casey and Richardson [9]	2008	The Impact of Fear on the Operation of Virtual Teams
Richardson et. al. [37]	2008	Having a Foot on Each Shore Bridging Global Software Development in the Case of SMEs
Boden et al. [4]	2009	Knowledge Management in Distributed Software Development Teams Does Culture Matter?
Casey [8]	2009	Leveraging or Exploiting Cultural Difference?
Dittrich and Giuffrida [15]	2011	Exploring the role of instant messaging in a global software development project
Shah et al. [41]	2011	Outsourced, offshored software- testing practice: Vendor-side experiences
Hattori et al. [22]	2012	A Qualitative User Study on Preemptive Conict Detection
Moe et al. [30]	2012	From offshore outsourcing to offshore insourcing: Three stories
Tamburri et al. [47]	2012	On the Nature of GSE Organizational Social Structures: an Empirical Study
Bass [3]	2013	Agile Method Tailoring in Distributed Enterprises: Product Owner Teams
Moe et al. [31]	2013	From offshore outsourcing to insourcing and partnerships: four failed outsourcing attempts
Nguyen-Duc and Cruzes [32]	2013	Coordination of software development teams across organizational boundary An exploratory study
Paasivaara et al. [34]	2013	Integrating Global Sites into the Lean and Agile Transformation at Ericsson
Shah and Harrold [40]	2013	Culture and Testing: What is the Relationship?
Zieris and Salinger [50]	2013	Doing Scrum Rather Than Being Agile: A Case Study on Actual Nearshoring Practices

members switched to English and continued the meeting. Danish team considered this behaviour impolite and it was negatively affecting the collaboration; the researcher, likewise, before visiting the Chinese team, was surprised by the behaviour, having observed the collaboration from the Danish side and having assimilated the Danish perspective. Thanks to the onsite visit and observing what was happening on the other side of the screen, before the meeting, and in the room during the meeting, and getting to know the participants, the researcher could understand the reasons for the odd behaviour. Chinese students were shy to talk a foreign language and uncomfortable with their English skills. During the meeting, Chinese members were clarifying to each other what was just said by Danish members, trying to formulate a proper reply and dealing with their embarrassment of talking a foreign language. Probably, it could have been possible to reach the same understanding if the collaboration was more mature and trust between teams was already at the stage to explicitly talk about challenges and misunderstandings. However, in the specific case of a novice team, only observations allowed to see what was not clear to the team members themselves and to understand human aspects that influenced the collaboration.

6.2. Example 2

Dittrich and Giuffrida [15] conducted an observational study, investigating the role of Instant Messaging (IM) in a GSD team distributed across Denmark and India. One researcher visited both sites and collected data through document analysis, semi-structured interviews and direct observations. Document analysis was performed to classify the usage of IM; interviews were used to check results with team members; and observations allowed gaining more insights about the usage of IM within the team. During an initial interview, Danish team members reported that IM chats were often distractive and uselessly interrupting their work: the impression of Danish managers was that Indian developers were misusing IM chats, frequently contacting Danish team members for unimportant reasons. For example, the content of the chat was often overlapping with information already available in the issue tracker system used. However, during the visit of the Indian site, the researcher observed that Indian members were very reflective before initiating an IM chat with remote members: prior to write to Danish team members, they were asking to co-located colleagues about solutions to their doubts, they were checking the issue tracker tool and when they finally decided to start an IM chat, they were typing the sentence several times in order to be sure to write a sentence that was meaningful and concise. Moreover, researchers observed that team members - both in India and in Denmark - were generally dutiful in the usage of the issue tracker system. After observing the attitude of team members toward the tools used, a detailed log analysis has been performed on IM logs and on the issue tracker system. The analysis of the IM logs showed that, despite this rigorous usage of the tool observed, in cases subjectively considered relevant, both Danish and Indian team members remarked the finalization of a task or its assignment to another member of the team also in a personal IM chat. An automatic notification was available in the issue tracker system through mail notifications, but team members did not use it. During a workshop dedicated to reflection on the observed practices, Danish team members put forward an interesting point: the hand over of tasks on IM chats allows maintaining the relationship with the remote developers, overcoming the lack of face-to-face communication due to geographical distance. According to the practitioners, the possibility of handing over tasks and supporting one another through small remarks and "smileys" is one of the reasons of not using the mail notification functionality of the issue tracker. On the contrary, an automatic notification would be annoying, because it would notify every single assignment, despite its contextual relevance. This behaviour is in contrast with considering that IM chats are misused, as reported in the initial interview with Danish Managers. Such social subtext of chats usage would not have been accessible through the sole usage of interviews. Observations - used with a supportive role - allowed to see that Indian developers were very reflective before initiating a chat with remote team members and that team members on both sites were properly using the issue tracker system, following detailed workflows. IM chats were "misused" by both sites, as visible from the log analysis. Team members themselves were not considering this perspective before being confronted with the researcher findings and the researchers would not have been able to ask without combining IM logs with on site observations - as to say: you can not ask what both researchers and practitioners do not suspect.

6.3. Example 3

Sharp et al [42] conducted an observational study of a partially-dispersed agile team. The project team consisted of one core team, an additional offshore testing team and a network of advisers (which was also globally dispersed). This study focused on the core team, which was made up of nine members dispersed over the US, Canada, Argentina, Holland and the UK. In the core team, many of the members worked on their own and were geographically dispersed from each other. They relied heavily on regular contact through the supplied infrastructure. All team meetings were supported by Lync, with screen sharing being used for demos or to share diagrams. Microsoft's OneNote and Team Foundation Server supported knowledge sharing. For this project a shared OneNote file with revision tracking that resided on a Skydrive was used, so that everybody could contribute. Team Foundation Server houses the source repository together with the product backlog and sprint backlogs with the current status of stories and bugs within the backlog. Skydrive was also used by the team as an additional project artefact repository. The project ran from July to November 2011 and the researcher observed the team for the whole of the project. The researcher attended two or three stand-ups a week and eight iteration planning meetings over the project, joined three triage sessions and several ad hoc conversations, visited USA and UK sites and obtained photographs of the Argentinean environment. She also had access to the team's OneNote notebook, which contained records of the teams retrospectives and many brainstorms and discussions. At the end of the project she was also given recordings of pairing sessions and other ad hoc meetings. The data collected included observation notes, screen captures of Lync conversations, still photographs, recordings of team conversations, pairing sessions and iteration planning meetings. Observation was the main research approach used in the study (central role). Informal interviews and questionnaires were also used to complement the observations and to check the observer's interpretations with the members. There were two clear benefits of using observation in this context: the researcher's experience of dispersed working was very close to the team members experiences; and the researcher could observe the team working without any embellishments which too often arise in interview or other self-reporting situations. The first benefit was specifically because all the teams substantial interactions had to be computer-mediated due to its dispersed nature. Although the researcher was not directly involved in producing software, the experience of knowledge sharing through the virtual environment, and of interacting with teammates over video conferencing and other media was shared. The empathy that this created was a significant bonus to the researcher when interpreting the data. The second benefit became clear when the team leader started to describe other activities that often happened during the project. For example having virtual presentations to the team, and having toys and food within one of the project sites. These were not present when the observations took place and no-one else referred to them. This means that they were not critical to supporting the teams work, and they were not generally seen as important by the team members. If we had relied only on self-reporting we may have taken a different interpretation of these activities and this information.

6.4. In Summary

The previous examples show that observations can be beneficial in GSD research for several reasons. Thanks to observations, it is possible to investigate human aspects that influence the collaboration and to explore aspects that would not be possible with the solely use of interviews or self-reporting. On the one hand, observing the remote team in the virtual world, where most of the collaboration happens, helps the researcher to share with team members the geographical distribution. On the other hand, travelling across sites allows observing on-site practices, limiting the biases of assuming just one perspective. Performing observations in GSD requires to consider how the team is distributed and to decide accordingly from where the team should be observed. This is further explored in the next section.

7. Performing Observational Studies in GSD

This section discusses how to deal with geographical distance, taking inspiration from the related works of ethnographers presented in Section 3.4. Ethnographic approaches involve a considerable amount of observations, thus they can provide suggestions on how to deal with observational studies. Before that, a taxonomy of possible geographical distribution of GSD teams is provided, based on the Systematic Mapping Study (SMS) performed in this paper and on the research experiences of the authors of the present paper.

7.1. Geographical Distribution of GSD teams

To the best of the authors knowledge, there is no common terminology for categorising geographical distribution of GSD teams. For example, the terms geographically distributed and geographically dispersed are used in an interchangeable way. Taking inspiration from a distinction proposed by Braithwaite and Joyce for agile teams [6], three kinds of global distribution are identified:

(a) *Globally Distributed teams.* Two or more teams are distributed around the Globe. The size of the teams and the number of teams can vary. Co-located physical collaboration happens in each site and global virtual collaboration happens across sites. This is the most widely investigated setup and many examples of this kind are available in the literature - e.g. [15], [13], [41], [30].

(b) *Globally Partially-dispersed teams.* One co-located team collaborates with one or more globally dispersed team members. There is at least one co-located team, but there can be multiple distributed teams. Co-located physical collaboration happens in the team, while global virtual collaboration happens among dispersed team members and between the core team and the disperse members - see e.g. the study of a partially-dispersed agile team by Sharp et al. [42].

(c) *Globally Dispersed teams.* Two or more team members are globally distributed and team members tend to be physically alone - thus dispersed. This is a common practice in Open Source (OSS) or for example when the access to most talented developers is needed. From the best of our knowledge, teams of this kind have been mainly investigated in OSS, see for example [43], [18].

Taking into account the possible ways teams are distributed is important for understanding from where researchers should observe them. This is further detailed in the next subsection.

7.2. From where Should I Observe?

Indipendently from the different possible ways teams can be distributed, it is important to observe team members where the intercations among them take place. However, it is generally not feasible to observe all sites at the same time: for example, in two globlly distributed teams, intercations occur at each physical site and in the virtual site. Thus, in the following, we provide an overview of four possible approaches that researchers should take into considerations when performing observational studies in GSD to deal with the different geographical distributions of team members presented in the previous paragraph. Examples from observational studies in GSD that adopt the different approaches are reported.

The first two approaches presented are based on Prikladnicki et al. assumption that "since it is often not possible to deploy several researchers to simultaneously observe work practices at different sites, processes are usually only observed from one teams perspective at a time" [36], thus Marcus multi-site ethnography [28] and strategically-situated ethnography [29], are adapted to GSD research. The third approach takes inspiration from virtual ethnography [24], while the latter considers a combination of the previous approaches taking inspiration from hybrid ethnography [25].

1. *Observe different physical sites in different moments.* This approach can be used mostly in globally distributed team and rarely for dispersed or partially dispersed teams. The researcher travels between different sites and can observe both the local collaboration in each co-located team and the virtual collaboration with remote teams, or dispersed team members, mediated by technology - e.g. [15], [30]. Basically the researcher "observes the global practices from a local perspective" [36]. When the researcher travels, he can observe the different physical sites and this has been reported as "usually the richest and most accurate way to understand the setup of the cooperation" [36]. Nevertheless, visiting the various physical sites augments the understanding of the co-located practices for each site, complementing the computer-mediated observations of the communication between sites. However, as Marcus [28] suggests for multi-site ethnography, it has to be taken into consideration that the researcher moves through and between field sites. Thus, his understanding of practice will be different: (1) the researcher can not observe the same event/activity from more than one site: travelling between sites, he can only observe other events/activities; (2) the researcher's knowledge about the team and about the project progresses during the study; thus, when the researcher travels to a different site, he deploys different perspectives and becomes aware of different issues. However, this would happen also in co-located settings: as the project progresses, the researcher gains more insights of the project; as a matter of fact, this is not necessarily a bad thing. Field notes, in this case, are the essential tool for the researcher to keep track of the progress. Examples of multi-site approach are the works by Giuffrida and Dittrich [15], [21] - see Examples 1 and 2 in Section 6 -, in which the researcher visited, respectively, both the Danish site and Chinese site, and both the Danish site and the Indian site. Thus, in both cases, local practices have been observed in different physical sites in different moments. Additionally, virtual collaboration was observed remotely from both sites, gaining the different perspectives of both remote sub-teams.

2. *Observe from one strategically-situated physical site.* When travelling in all sites is not possible due to cost, time, dimension of the project, company constraints, or when teams are partially-dispersed, the researcher can observe the collaboration of the distributed teams, as well as the collaboration of dispersed team members, being strategically-situated in one or more physical sites. This approach takes inspiration from strategically-situated ethnography [29], and relies on the fact that participants themselves are also trying to manage scale and distribution from a single site. The researcher identifies the key sites to visit or events to attend and further observations of the remote team are generally computer-mediated, as it is the collaboration between sites. For example, Shah et al. visited only the offshore team in India and gained insights also of the client teams in USA and UK [41]. The project of Patil et al. [35] involved four physical sites in USA and one in India: the researcher visited a total of four sites. One site in USA was not visited but two members from this site were interviewed while they were visiting another USA site. It is also possible to observe dispersed teams when team members meet in a single physical site for occasional meetings, as it happens in OSS - see e.g. [43]).

3. *Virtually observe the physical sites.* Sometimes travelling is not possible, or, as it happens in a dispersed team, it is not feasible to visit all locations because team members themselves are physically dispersed. In this case, virtual is the space in which the team acts and here the observations should take place. Since all team members interact only online, also the researcher can experience the same way of interaction. In GSD teams, virtual observations implies e.g. attending meetings, observing group multi chats, being involved in coding activities as a participant observer, without being co-located with any member of the team. The researcher sits in his office and observes the computer-mediated interaction as any member of the team. For example, Sharp et al. [42] - see Examples 4 in Section 6 - combined virtual observations with strategically-situated observations for investigating a partially-dispersed agile team. The researcher observed all team members remotely, with the purpose of understanding team members perspective of "having virtual colleagues and virtual meetings". Virtual observations have been also used to identify the strategically relevant sites that have been subsequently visited.

4. *Combine Research Approaches.* While the choice among multi site observations thus observing different physical sites in different moments - and strategically-situated observations usually depends on the size and on the setup of the project, in GSD, virtual observations are always performed, either as the solely way of observing the team or in combination with one of the other two approaches. Virtual observations are essential to gain an understanding of the remote collaboration in GSD teams; when they are used in combination with physical observations, we can define it as a hybrid approach [25], that allows investigation of the hybrid world in which software team members are immersed. In particular, the digital traces - and the digital world that support the physical work of the team - are investigated as a complementary field site for the (multi) physical one(s). Essentially, all examples reported in Section 6 involve a combination of physical and virtual observations, thus they all adopt a hybrid approach - see Table 5.

Table 5: Approaches adopted by the four examples reported in Section 6.

Ex	Authors	Role of observation	Multi-site observations	Strategically situated observations	Virtual observations	Hybrid approach
1	Giuffrida and Dittrich [21]	supporting	X		X	X
2	Dittrich and Giuffrida [15]	supporting	X		X	X
3	Sharp et al. [42]	central		X	X	X

8. Conclusions

Qualitative research methods have been found to be critical for grasping and understanding the phenomena in which humans play a role [45]. In particular, observational studies allow investigating cultural, socio-technical and human aspects, which are essential part of Global Software Development (GSD) activities. In this paper, we performed a Systematic Mapping Study to show which research methods are currently adopted to investigate the GSD field. Moreover, we categorized two possible usages of observations - as a central or as a supporting method of investigation -, highlighting the importance of direct observations for triangulating results. We also presented some examples deriving from our research experience, highlighting the advantages of performing observational studies in GSD. We finally described the challenges that researchers can encounter when performing observations in distributed settings and we provided suggestions on how to overcome those challenges. The main contributions of the present work can be summarised as:

1. In GSD research, observational studies are not widely adopted, GSD research is mostly carried on through surveys and interviews;

2. Observations are always used in combination with other research methods, such as interviews and document analysis. Observations can be used as the central data collection technique, or they can support other research methods.

3. Observations help to catch human aspects involved in GSD projects;

4. Four approaches are proposed about how to perform observational studies dealing with geographical distributions of team members (multi-site, strategically-situated, virtual observations, hybrid approach).

Based on the mapping of the methods - showing the limited usage of observational methods - and on the exemplar studies - showing the role of observations - we argue for the need of more observational studies in GSD, rather than relying on interviews and surveys as the primary method of investigation, as often researchers can not ask what they do not suspect.

References

[1] Aranda, J., Venolia, G.: The secret life of bugs: Going past the errors and omissions in software repositories. In: Proceedings of the 31st International Conference on Software Engineering, pp. 298–308. IEEE Computer Society (2009)

[2] Basili, V.R.: The role of experimentation in software engineering: past, current, and future. In: Proceedings of the 18th international conference on Software engineering, pp. 442–449. IEEE Computer Society (1996)

[3] Bass, J.M.: Agile method tailoring in distributed enterprises: Product owner teams. In: Global Software Engineering (ICGSE), 2013 IEEE 8th International Conference on, pp. 154–163. IEEE (2013)

[4] Boden, A., Avram, G., Bannon, L., Wulf, V.: Knowledge management in distributed software development teams-does culture matter? In: Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on, pp. 18–27. IEEE (2009)

[5] Boellstorff, T., George, E., Nardi, B., Pearce, C., Taylor, T.: Ethnography and Virtual Worlds: A Handbook of Method. Princeton University Press (2012)

[6] Braithwaite, K., Joyce, T.: Xp expanded: Distributed extreme programming. Extreme programming and agile processes in software engineering pp. 1524–1526 (2005)

[7] Bugde, S., Nagappan, N., Rajamani, S., Ramalingam, G.: Global software servicing: Observational experiences at microsoft. In: Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on, pp. 182–191. IEEE (2008)

[8] Casey, V.: Leveraging or exploiting cultural difference? In: Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on, pp. 8–17. IEEE (2009)

[9] Casey, V., Richardson, I.: The impact of fear on the operation of virtual teams. In: Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on, pp. 163–172. IEEE (2008)

[10] Cataldo, M., Bass, M., Herbsleb, J.D., Bass, L.: On coordination mechanisms in global software development. In: Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on, pp. 71–80. IEEE (2007)

[11] Conchúir, E.Ó., Ågerfalk, P.J., Olsson, H.H., Fitzgerald, B.: Global software development: where are the benefits? Communications of the ACM **52**(8), 127–131 (2009)

[12] Curtis, B., Krasner, H., Iscoe, N.: A field study of the software design process for large systems. Communications of the ACM **31**(11), 1268–1287 (1988)

[13] Damian, D., Izquierdo, L., Singer, J., Kwan, I.: Awareness in the wild: Why communication breakdowns occur. In: Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on, pp. 81–90. IEEE (2007)

[14] Damian, D.E., Zowghi, D.: The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization. In: Requirements Engineering, 2002. Proceedings. IEEE Joint International Conference on, pp. 319–328. IEEE (2002)

[15] Dittrich, Y., Giuffrida, R.: Exploring the role of instant messaging in a global software development project. In: Global Software Engineering (ICGSE), 2011 6th IEEE International Conference on, pp. 103–112. IEEE (2011)

[16] Dittrich, Y., John, M., Singer, J., Tessem, B.: Editorial: for the special issue on qualitative software engineering research. Information and software technology **49**(6), 531–539 (2007)

[17] Easterbrook, S., Singer, J., Storey, M.A., Damian, D.: Selecting empirical methods for software engineering research. Guide to advanced empirical software engineering pp. 285–311 (2008)

[18] Elliott, M.S., Scacchi, W.: Free software developers as an occupational community: resolving conflicts and fostering collaboration. In: Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work, pp. 21–30. ACM (2003)

[19] Estler, H.C., Nordio, M., Furia, C., Meyer, B., Schneider, J.: Agile vs. structured distributed software development: A case study. In: Global Software Engineering (ICGSE), 2012 IEEE Seventh International Conference on, pp. 11 –20 (2012). DOI 10.1109/ICGSE.2012.22

[20] Giuffrida, R., Dittrich, Y.: Empirical studies on the use of social software in global software development-a systematic mapping study. Information and Software Technology (2013)

[21] Giuffrida, R., Dittrich, Y.: How social software supports cooperative practices in a globally distributed software project. In: ICSE Workshop on Cooperative and Human Aspects of Software Engineering. ACM (2014)

[22] Hattori, L., Lanza, M., D'Ambros, M.: A qualitative user study on preemptive conflict detection. In: Global Software Engineering (ICGSE), 2012 IEEE Seventh International Conference on, pp. 159–163. IEEE (2012)

[23] Hess, E.R., Audy, J.L.N.: Ftsproc: A process to alleviate the challenges of projects that use the follow-the-sun strategy. In: Global Software Engineering (ICGSE), 2012 IEEE Seventh International Conference on, pp. 56–64. IEEE (2012)

[24] Hine, C.: Virtual ethnography. Sage (2000)

[25] Jordan, B.: Blurring boundaries: The" real" and the" virtual" in hybrid spaces. Human Organization **68**(2), 181–193 (2009)

[26] Korkala, M., Abrahamsson, P.: Communication in distributed agile development: A case study. In: Software Engineering and Advanced Applications, 2007. 33rd EUROMICRO Conference on, pp. 203–210. IEEE (2007)

[27] Lanubile, F.: Collaboration in distributed software development. In: Software Engineering, pp. 174–193. Springer (2009)

[28] Marcus, G.E.: Ethnography in/of the world system: the emergence of multi-sited ethnography. Annual review of anthropology pp. 95–117 (1995)

[29] Marcus, G.E.: Ethnography through thick and thin. Princeton University Press (1998)

[30] Moe, N.B., Hanssen, G.K., et al.: From offshore outsourcing to offshore insourcing: Three stories. In: Global Software Engineering (ICGSE), 2012 IEEE Seventh International Conference on, pp. 1–10. IEEE (2012)

[31] Moe, N.B., Šmite, D., Hanssen, G.K., Barney, H.: From offshore outsourcing to insourcing and partnerships: four failed outsourcing attempts. Empirical Software Engineering pp. 1–34 (2013)

[32] Nguyen-Duc, A., Cruzes, D.S.: Coordination of software development teams across organizational boundary–an exploratory study. In: Global Software Engineering (ICGSE), 2013 IEEE 8th International Conference on, pp. 216–225. IEEE (2013)

[33] Nicolini, D., Gherardi, S., Yanow, D.: Knowing in organizations: A practice-based approach. ME Sharpe (2003)

[34] Paasivaara, M., Lassenius, C., Heikkila, V.T., Dikert, K., Engblom, C.: Integrating global sites into the lean and agile transformation at ericsson. In: Global Software Engineering (ICGSE), 2013 IEEE 8th International Conference on, pp. 134–143. IEEE (2013)

[35] Patil, S., Kobsa, A., John, A., Seligmann, D.: Methodological reflections on a field study of a globally distributed software project. Information and Software Technology **53**(9), 969–980 (2011)

[36] Prikladnicki, R., Boden, A., Avram, G., Souza, C., Wulf, V.: Data collection in global software engineering research: learning from past experience. Empirical Software Engineering pp. 1–35 (2013). DOI 10.1007/s10664-012-9240-x. URL `http://dx.doi.org/10.1007/s10664-012-9240-x`

[37] Richardson, I., Avram, G., Deshpande, S., Casey, V.: Having a foot on each shore-bridging global software development in the case of smes. In: Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on, pp. 13–22. IEEE (2008)

[38] Robson, C.: Real world research: a resource for social scientists and practitioner-researchers, vol. 2. Blackwell Oxford (2002)

[39] Seaman, C.B.: Qualitative methods in empirical studies of software engineering. Software Engineering, IEEE Transactions on **25**(4), 557–572 (1999)

[40] Shah, H., Harrold, M.J.: Culture and testing: What is the relationship? In: Global Software Engineering (ICGSE), 2013 IEEE 8th International Conference on, pp. 51–60. IEEE (2013)

[41] Shah, H., Sinha, S., Harrold, M.J.: Outsourced, offshored software-testing practice: Vendor-side experiences. In: Global Software Engineering (ICGSE), 2011 6th IEEE International Conference on, pp. 131–140. IEEE (2011)

[42] Sharp, H., Giuffrida, R., Melnik, G.: Information flow within a dispersed agile team: A distributed cognition perspective. Agile Processes in Software Engineering and Extreme Programming pp. 62–76 (2012)

[43] Sigfridsson, A., Sheehan, A.: On qualitative methodologies and dispersed communities: Reflections on the process of investigating an open source community. Information and Software Technology **53**(9), 981–993 (2011)

[44] Sim, S.E., Singer, J., Storey, M.A.: Beg, borrow, or steal: Using multidisciplinary approaches in empirical software engineering research. Empirical Software Engineering **6**(1), 85–93 (2001)

[45] Sjoberg, D.I., Dyba, T., Jorgensen, M.: The future of empirical methods in software engineering research. In: Future of Software Engineering, 2007. FOSE'07, pp. 358–378. IEEE (2007)

[46] Šmite, D., Wohlin, C., Gorschek, T., Feldt, R.: Empirical evidence in global software engineering: a systematic review. Empirical software engineering **15**(1), 91–118 (2010)

[47] Tamburri, D.A., Lago, P., Vliet, H.V., Di Nitto, E.: On the nature of gse organizational social structures: An empirical study. In: Global Software Engineering (ICGSE), 2012 IEEE Seventh International Conference on, pp. 114–123. IEEE (2012)

[48] Taxén, L.: An integration centric approach for the coordination of distributed software development projects. Information and software technology **48**(9), 767–780 (2006)

[49] Wohlin, C., Runeson, P., Host, M., Ohlsson, C., Regnell, B., Wesslén, A.: Experimentation in software engineering: an introduction (2000)

[50] Zieris, F., Salinger, S.: Doing scrum rather than being agile: A case study on actual nearshoring practices. In: Global Software Engineering (ICGSE), 2013 IEEE 8th International Conference on, pp. 144–153. IEEE (2013)

Exploring the Role of Instant Messaging in a Global Software Development Project

Yvonne Dittrich
Software Development Group
IT University of Copenhagen
Copenhagen, Denmark
ydi@itu.dk

Rosalba Giuffrida
Software Development Group
IT University of Copenhagen
Copenhagen, Denmark
rogi@itu.dk

Abstract— **Communication plays a vital role in software development projects. Globally distributed teams use a mix of different communication channels to get the work done. In this paper, we report on an empirical study of a team distributed across Denmark and India. This paper explores the integration of formal documentation, bug-tracking systems and email with informal communication on Instant Messaging (IM), screen sharing, and audio conversations. Whenever overlap times occur, informal communication can take place at the same time in different sites, and it can effectively complement formal documentation. Our analysis provides an indication that IM can play a special role in such socio-technical communication systems: IM acts as a real time glue between different channels. The communication through IM also provides a means to build trust and social relationships with co-workers.**

Instant messaging, skype, social software, informal communication, ethnographic research.

I. INTRODUCTION

Software development teams are used to working with many different tools for developing software, versioning it, scheduling work, managing requirements and test cases, and sharing knowledge with other team members. There are several tools available for these activities, such as: integrated development environments, versioning systems, bug-trackers, test management tools, word documents and email. These tools can be considered as channels in which formal communication takes place among virtual teams.

Formal communication is not always sufficient for conducting successful software projects. In co-located teams, planned or impromptu face-to-face meetings can easily support the development process, complementing the formal knowledge being shared through the use of tools [1]. In distributed teams, informal communication cannot take place in person very easily, and thus, other kinds of unstructured channels are used in Global Software Development (GSD), e.g. IM, voice calls, and screen sharing. Turner et al. [34] have explored the workplace communication ecology, investigating why particular tools in a user's ecology are selected, and how the tools are used in combination.

In this study, we investigate the concept of communication ecology in globally distributed *teams* in order to answer the following question:

> How do software distributed teams manage communication using different channels and how do they integrate the communication across different channels?

In our analysis, we refer to the concept of coordination mechanisms proposed by Schmidt and Simone [25]: coordination mechanisms consist of a coordinative social protocol sometimes imprinted on, and supported by, distinct artifacts. They stipulate and mediate the articulation of cooperative work so as to reduce the complexity of the articulation work of that arrangement. A bug tracker, for example, collects bug reports and supports structured asynchronous communication that sustains the debugging process. Skype™ is an example of less structured communication, nearly a synchronuous communication channel. Coordination mechanisms [25] can use structured and unstructured communication channels. For unstructured channels, however, it is necessary to re-negotiate coordination mechanisms [25].

This paper reports a qualitative study on collaboration in a distributed software development team. We observe that IM plays a crucial role in supporting daily work and coordination across sites: IM acts as the glue between different channels available in the communication ecology [34] of the virtual team.

We believe that our study can help to bring about a better understanding of how formal and informal communication can be combined in GSD, as well as to understand the crucial role of IM for coordination, collaboration and socialization in software distributed teams.

II. RELATED LITERATURE

The fundamental problem of GSD is that many of the mechanisms that function to coordinate the work in a co-located setting are absent or disrupted in a distributed project [14]. Bruegge et al. [3] have proposed a framework for enabling informal collaboration in global software development, underlying the crucial role that informal mechanisms play in distributed software teams.

While traditionally the main media for informal communication in distributed teams have been email, phone, and video conferencing systems [1] [11], nowadays a lot of communication takes place in the so-called Social Software (SoSo) [10]. SoSo can be seen as a repository of non-structured knowledge where informal communication and knowledge sharing take place. SoSo includes a wide variety of tools such as: IM, Internet forums, mailing lists, blogs, wikis, social network sites, social bookmarking, social libraries, and virtual worlds. The main challenge from a Software Engineering perspective is that informal communication is not structured by itself. While Software Engineering usually focuses on structured documentation and communication, knowledge shared through SoSo is unstructured. As high-expertise practitioners involved in distributed team projects share their knowledge both through formal documentation and informal situated meanings, SoSo should be taken into account as a complementary link.

Many research projecrs in the CHI, CSCW and IS communities have already investigated the applications of SoSo in corporate contexts [32] [33] [35]. Despite the increase of Web 2.0 tools, research on the use of SoSo in workplaces still focuses mainly on IM. One of the first papers on IM for collaborative work was written in 2000 by Nardi et al. [20]; the authors document the flexibility and expressivity of IM for various informal communication tasks in the workplace. In 2010, Ou et al. [22] demonstrated that IM can empower teams at work. In the last ten years, a lot of research has been conducted on the use of IM in the workplace, especially in distributed settings; we have identified three main empirical research directions: evaluation of self-made IM tools, laboratories studies with students, and analysis of practices adopted in open source communities.

Hubbub and WebWho are two examples of IM tools developed in the early 2000s. Issacs et al. [17] logged thousands of workplace conversations among users of their Hubbub system, and evaluated the nature and functions of the conversations. They found that the primary use of Hubbub was for work conversations; secondary uses were for simple, single purpose interactions and for scheduling or coordination. Studies conducted with WebWho, a web based awareness system that visualizes where people are located in a large university computer lab, show that students use the IM to support collaborative work and coordinate social activities, and use it extensively for playful behavior [27].

The laboratory studies conducted by Setlock et al. [28] mainly underlined the importance of cultural aspects in computer mediated communication. Other ad-hoc studies have investigated the effects on interpersonal perception [6] or the effects on cooperation, persuasion, and deception [2] or comparisons between chat and audio in media rich environments [26]. Also, in OSS settings, the use of IM has been researched extensively. Elliot [9] indicates the importance of recorded logs of IM for resolving conflicts in virtual work communities. The use of persistent online IRC logs and mailing list archives serves to tie the virtual work group together, contributing to conflict resolution and, at the same time, reinforcing the beliefs in free software and freedom of choice, as well as the more tacit values of cooperative work and community building. Gutwin et al. [12] affirm that text chat is suitable for ad-hoc communication and ‘overhearing’ of informal and work-related discussions. We notice that little empirical research on the use of IM in GSD has been reported, and thus believe it is appropriate to investigate this aspect in real world settings.

Niinimaki and Lassenius [21] have studied the use of commercial-grade, widely adopted and acknowledged IM tools in commercial GSD projects, through a multiple qualitative case study. They reported on successful use of IM in global distributed projects. Thissen et al. [31] describe the role of different communication tools for distributed software development teams, and Turner et al. investigate [34] how people set their own communication ecology in the workplace according to communication needs. Combining these two works, we conclude that in GSD a team’s communication ecology of different tools used for different purposes is evident.

Based on ethnographic research, we emphasize the central role of IM as glue between different channels mediating the collaboration and assemblies of boundary objects [4] [19]. We analyze the role of IM by identifying and investigating different dimensions of chat usage. Handel and Herbsleb [13] have already built a classification of chats, defining five categories: availability, non-work topics, work, greeting, and humor. In our analysis, we observe that, probably because today IM is more integrated in everyday work practices, all chats are work related and several dimensions are present at the same time.

III. Case Description and method

DHI is an independent, international, consulting and research organization. The company develops and uses high-end hydraulic simulation software. We have investigated how informal communication takes place inside the World Bank Project (WB-Project), which has a considerable amount of ongoing software development. This is a global distributed project: five members are in Copenhagen, Denmark; seven members in Delhi, India and one Project Area Manager in Portland, USA. The Danish team is composed of one project manager and four Project Area Managers (PAM), the Indian team consists of five developers, one team leader, and one tester. Each PAM is responsible for one specific part of the software and collaborates with one or two Indian developers. Project Area Managers are both domain experts of hydraulic engineering and software developers. They take care of the management of the project, requirement specifications, quality assurance processes as well as the design of the software and its implementation. We observed the team while working on the development of a Decision Support System (DSS) for water management in the Nile Basin. The observations took place during the final part of the development process of the first release of the system mainly during the testing phase of the same release. The project was successful, and the team is now developing the second release of the software.

The ethnographic empirical research took place both in Copenhagen and in Delhi. Researchers collected field material and observed the team for four months from

Denmark. In order to have a better understanding of the practices, researchers also spent two weeks in India.

For this study, researchers collected data through different qualitative empirical techniques: participant observation, semi-structured interviews, member checking and document analysis. In order to carefully track the investigation, researchers taped and transcribed meetings and interviews, they applied interaction analysis on IM logs, and they kept a research diary during the entire project. Workshops were organized to summarize the outcomes and to support researchers and practitioners in reflecting together on observations. By participating in the daily routines, it was possible to observe how collaboration took place between different participants of the development project. Interviews gave the opportunity to clarify uncertainties and for interviewers to ask about specific issues in a deeper way.

The analysis of the data was initially performed during the observation period through several iteration of analysis of the field material. Through these iterations, it was possible to reflect on the observed practices and to re-adapt the data collection. Once the observation period ended, researchers re-analyzed the field material in order to summarize the findings.

By using multiple ways of collecting data and combining different kinds of methods, it has been possible to triangulate the findings [24]. The goal of the empirical part of the research is to understand and support software development from a shop floor perspective, as proposed by Dittrich et al. [8] in the Cooperative Method Development (CMD), which is an approach for investigating social and cooperative aspects of software development. With the support of the researchers, team members had the opportunity to reflect on their own practices and to understand difficulties that they encountered during everyday collaboration.

IV. Analysis

In this section, we describe the WB-Project team, outline which tools the team used, determine how they were used, and present the incident workflow. Finally, we describe the different dimensions of the use of IM that play a central role in the communication ecology of the team.

A. The World Bank project team

The working day of the team starts in India. In the Delhi office, developers arrive starting from 9 a.m., Indian Standard Time (IST) or 4:30 a.m., Central European Time (CET). They are gathered all together in an open space. A big table is shared by all team members sitting on the two long sides of the table. The table is split into personal workspaces by small dividers. The morning is rather quiet, as developers finish their work of the previous day or start newly assigned tasks. Sometimes they move to another workspace to communicate with other team members.

Collaboration across sites takes place during four to six overlapping working hours for different purposes: clarifying requirements, debugging pieces of software, and coordinating work tasks. Real time collaboration and communication across sites usually starts through Skype™ instant messaging, but a whole set of channels is used by team members, depending on the specific task that they have to perform.

At the time of the late Indian morning, Danish managers wake up with some of them starting to work around 8 a.m., CET or 12:30 IST, often from home. When Danish team members appear online on Skype™, Indian developers start asking them questions, or a Danish PAM might call Indian developers to organize his working day. Some real time collaboration can take place in this short slot before the Indian lunch that takes place at 1 p.m., IST. Collaboration is usually interrupted at the Indian side for lunch hour, as all team members take lunch together.

While the Indian team is having lunch, the Danish team members reach the office. The five team members are spread in four two-person-offices in two different floors. After Indian members come back from lunch, at 10 a.m., CET, Danish team members usually have a meeting. During this routine meeting, PAMs plan future activities, prioritize tasks and assign incidents and tasks to team members. The meeting usually lasts about one hour. The time after the meeting, from 11 a.m. to 12 a.m., is normally one of the most intense times for communication across sites.

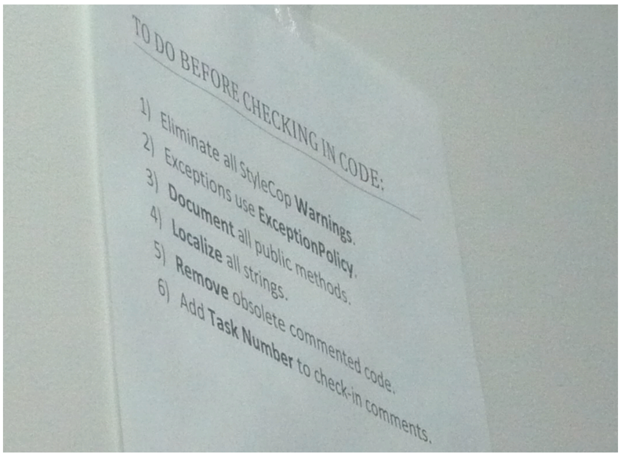
About 12 noon, CET, the Danes have lunch. Their lunchtime is flexible and depends on their current activities, as they usually do not share lunch. When the Danish team is back from lunch, the working day in India is close to ending. The last slot for real time communication and collaboration is from 5 p.m. to 6 p.m., IST. The rest of the Danish afternoon is quieter when local collaboration and meetings take place.

B. Different tools used

The main structured collaboration support is *Spira*™, a Test Management tool from Inflectra Corporation. It is used by the team as an issue tracking system. All development activities are tracked in this system: the description of features to develop, reporting of incidents, assignment of tasks, and description of test cases. Spira™ is a test case tool rather than a project management tool, but it is used by the team to manage tasks of the team members. The current version has some limitations for user management and there are few bugs in the activity reporting functionality. Spira™ automatically assigns an "incident number" (IN) to all defects, test cases (TC) and requirements (REQ). These numbers are used by the team in all tools to link the Spira™ entries with chat, mail, documentation and source code: for example, every time developers check in code in the source repository, they also report the related number of Spira™ (See Figure 1).

Skype™ is the team's main tool for collaboration and team members are supposed to have it switched on when they are at work. Team members use different Skype™ channels for synchronous or nearly synchronous collaboration: written IM, audio, and screen sharing. Usually, one-to-one communication and collaboration take place through Skype™. Through the *contact list*, team members get an overview of the availability of remote colleagues. The availability information in the contact list provides a sense of connectediness, bringing people together

[20]: if colleagues are online during working hours, team members assume that they are available for chat. In the team, both Indian and Danish members do not make much use of broadcasting information through the status information of the contact list [29], and there are no policies about it. Only some of the PAMs indicate whether they are *@dhi.dk*, *@Delhi* or *in Portland* as they travel much of the time.



TO DO BEFORE CHECKING IN CODE:

1) Eliminate all StyleCop **Warnings**.
2) Exceptions use **ExceptionPolicy**.
3) **Document** all public methods.
4) **Localize** all strings.
5) **Remove** obsolete commented code.
6) Add **Task Number** to check-in comments.

Figure 1. Todo before checking the code

The most used Skype™ functionality is IM. Indian developers state that they contact the Danish team through IM *"whenever immediate feedback is required or due to urgent or high priority issues."* In other cases, they prefer using mails. Danish team members use IM for checking what developers are working on, as well as for giving clarification and feedback. The function of chat is further analyzed in Section D of the analysis.

Voice over IP allows team members to talk together and is replacing traditional phone connections. Due to the fact that contact availability can be seen on the contact list, calls happen only if the person is online on Skype™. Moreover, the call is usually negotiated beforehand, through a chat on IM, so that each member of the team can check the actual communication availability of the colleague.

While talking to each other, team members often *share screen.* Referring to the same indexical space, team members can discuss code, software behavior, or incident description. Screen sharing is used for debugging, for code/solution review and also for short session of pair programming. We observed that developers usually show their screen to PAMs, rather than the other way round. Skype™, however, does not support remote control of the desktop. Only one side can point and manipulate the software and tools, a limitaton which leads to delays and misunderstandings.

When team members need to control the remote desktop, they use a proprietary tool developed by DHI that is mainly used for remote presentations for customers. This tool requires preparation by the person who wants to share the screen and the sending of the code to the remote colleague. The decision to use this facility is usually made during a call, when the Skype™ screen sharing turns out to be insufficient. Despite the fact that *video* is one of the core features of Skype™, one important factor that we observed is that no one in the team uses video. In the WB-project the team does not miss face-to-face contact at all. The following citation is an example of what both the Indian developers and Danish team members reported:

> *I don't have a camera but I can use the external one if I want. I don't feel we need to use video… Actually we use video: we share our screen when necessary.*

Together with Spira™ and Skype™, team members also make use of traditional *email*. Emails are exchanged among all team members, mainly whenthere will be a time delay for fining a solution or when it is necessary to involve more people in the discussion. Especially when the input from the US team member is necessary, the communication takes place by using emails as there is only little overlap in working hours with Portland. Email is also used to broadcast information, to formalize decisions, and to ask for more formal response.

Both *Word* and *PDF documentation* are used to formalize the planning and the execution of the WB project. Project Organization and Project Life Cycle are formally defined in a document with guidelines for the Software Development Project. A document incorporates a short, step-by-step guide on how Spira™ should be used by the team to manage software development projects. These documents are shared through mail or are uploaded in Spira™. Interestingly, in these documents, Skype™ is not mentioned as a tool for supporting the software development process.

C. The incident work flow

In the "Project handbook" document, there is a diagram that formally describes the workflow of a defect (see Figure 2). The defined life cycle is roughly followed. The process is clearly defined: a defect that has been registered - in Spira™ - has to be approved by the PAM, who can reject it or evaluate it. During the morning meeting, Danish managers – Change Control Board (CCB) - usually approve and plan the incident, assigning it to a developer through Spira™. When the developer starts working on the defect, he changes its status as "in progress," and when he finishes working on that, he marks it as "completed." He then assigns the incident to the PAM, who can test and verify it. This process, defined through the diagram, is strictly mapped in the Spira™ system, and the status of all defects is also structured in the tool. Apart from defect definition and design comments in the Spira™ through the whole workflow, a lot of informal communication takes place in order to clarify the formal specification. Mails, voice calls, screen sharing and IM chats enrich the Spira™ content.

Following the lifecycle of a specific incident, we identify three phases in which informal communication usually takes place across sites: when the defect is assigned to the PAM, when the developer works on the defect, and when the defect is in status "completed."

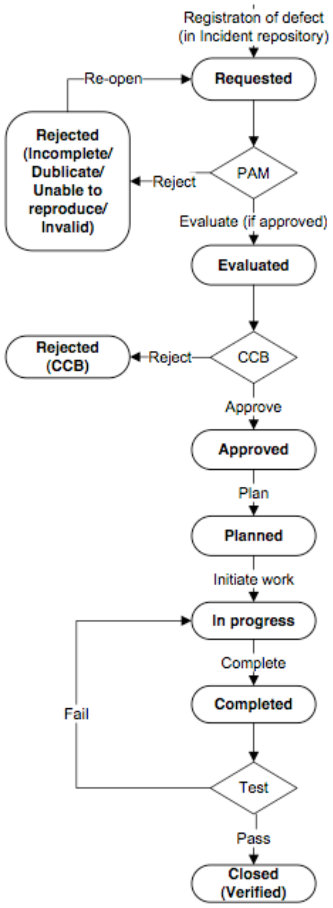


Figure 2. The Defect workflow

In the first phase, if a tester or a developer assigns the defect to a PAM, the steps to reproduce the incident described in Spira™ could be insufficient for the PAM to reproduce it. In this case, he chats with the Indian member through IM, or he may ask to start a screen sharing session in order to better understand the problem. If a developer is working on the resolution of an incident, and has doubts on how to implement the solution described in Spira™, he contacts the PAM to discuss the technical issues through IM, audio, or screen sharing. Finally, when an incident is closed, it can happen that the implemented solution does not completely solve the issue, or the solution can interfere with other parts of the software. In the first case, the PAM uses chats and audio to talk directly with the developer. In the second case a mail is sent to all the PAMs and developers involved, in order to inform all members about the problem and to find a shared solution. In other words, the resolution of incidents takes place not only through Spira™ and through the source code repository, but deploys a whole range of structured and un-structured channels. The informal communication and collaboration makes the formal process work.

D. Dimensions of the use of IM

In this section, we focus on informal communication through IM and we analyze and classify the chat content. The different dimensions that we consider for our analysis are: coordination, collaboration, awareness, and socialization. Each chat can normally be classified under a predominant dimension. Interestingly, often more than one dimension can come into play at the same time and in the same piece of chat. We here briefly describe the four dimensions, before we show how they become visible in the different chats.

The *coordination dimension* is the most prominent one. Although tasks and incidents are assigned to team members through Spira™, it often happens that people coordinate informally on what to do next or on managing the lifecycle of an incident or a task. Chats are another consideration for negotiating a call as a coordination chat. Usually team members do not call directly people on their contact list: the availability of the counterpart is checked through a written chat. The call is often required when people need to discuss a specific task or incident or want to show/check solutions implemented.

When technical issues are discussed the *collaboration dimension* becomes visible. These chats are usually easy-to-resolve issues that do not require much discussion. We observed that collaboration chats usually last for no more than 10 minutes; on average, they last 7 minutes. When the topic is too complex for a written chat, the conversation started in IM usually switches to a call. Moreover, if issues are considered to have a wider implication, developers or PAMs may start a discussion on mail.

For the *awareness dimension* chats are used: for example, Indian developers may mention what they are working on in order to avoid conflict. On the other hand, Danish managers check explicitly on what people are working on to plan future activities. This information is provided by Spira™, but often what the official tool shows will not necessarily correspond to what people are really working on at the moment. This is due to technical limitations of the tool and also because of remiss reporting of activities. Many of the coordination chats provide awareness information to the cooperation partner.

The *socialization dimension* is visible both in dedicated social chats and as sub-text in other chats. They are usually talks unrelated to code regarding working conditions, commuting problems or personal topics. These chats can be longer than coordination and collaboration ones but, since they are low priority conversations, usually they are going on while people are also working on some other tasks. The socialization dimension is sometimes shown as an addendum to a coordination or awareness chat, e.g., as positive feedback for the good job. This kind of chats is useful for team building and mutual appreciation, as well as for motivation. During many collaboration and coordination chats, implicit exchange of awareness and socialization as sub-text takes place and it is very interesting to see how these aspects are interwined together with coordination and collaboration activities.

1) *The Coordination Dimension*

Despite the fact that all team members have to assign tasks or incidents to other colleagues on Spira™, we observed that manu *coordination chats* start from the Indian side to inform the task/incident changed status, and is assigned to the counterpart.

```
[14:07:17] IND: 974 is yours now for verification (gap filler scrollbars)
[14:09:47] DK: thanks
```

(1)

```
[13:23:48] IND: yes 699 and 689 done. 689 deviates a bit from DK2 solution. So does 699. please review them.
[13:24:05] DK: alright
```

(2)

As we can see in the previous example, coordination chats give implicit awareness to Danish PAMs about what developers are working on, and about what they are supposed to do next, such as checking the solution implemented.

We observe frequent coordination from both sides in very close exchange of assignments, as in pair programming settings:

```
[14:07:17] IND: 974 is yours now for verification (gap filler scrollbars)
[14:09:47] DK: thanks
[11:58:39] IND: 1008
[12:07:32] DK: all yours
[12:57:18] IND: your now.
[15:58:02] DK: 847 is yours
```

(3)

It happens also that developers ask what the tasks are that they are supposed to do, as in the following example:

```
[08:24:46] IND: any incident you have planned for me...
[08:24:57] IND: to fix
[08:28:15] DK: look in SPIRA for planned - take from the top with respect to priority
```

(4)

Similarly, chat is used to communicate about sending/receiving mail:

```
[20:21:23] DK1: hi
[20:26:27] DK2: writing ou an email now :)
[20:26:55] DK1: just replied to it
[20:30:14] DK2: no...another one :)
[20:30:17] DK2: just sent it now
```

(5)

```
[07:28:33] IND: Hi DK...I have sent a mail..pls check
[07:29:57] DK: ok
```

(6)

Also in this case the coordination dimension is interwined with the socialization one. We can observed a need to give attention to specific artifacts that are considered crucial for developing the work.

Negotiation chats look pretty similar, whether started from the Indian or Danish side; however, it is more typical that they are started by the Indian developers. Two examples, from both sides, are reported below.

```
[10:51:52] IND: hi..
[10:59:05] DK: hi
[10:59:49] IND: can we discuss in811
[11:09:58] DK: sure
[11:10:58] ** Call to IND, no answer. **
[11:11:16] DK: I'll call again in 10 minutes - need to talk to somebody
[11:14:22] IND: ok, sorry i went to IND2 as he had some problems in the build
[11:15:28] DK: let's talk now - I never got away from the keyboard :)
[11:16:13] IND: he he.. ok :)
[11:16:47] ** Call from IND, duration 10:42. **
```

(7)

```
[09:40:32] IND: Hi DK..
[09:40:59] IND: I have implemented […]
[09:41:38] IND: the implementation is working fine
[09:41:58] IND: do u want me to check-in the code and test on ur system??
[10:23:21] DK: maybe you could show it to me ?
[10:23:36] DK: I'll call when DK2 is ready
[10:24:07] IND: ok.call me once you guys are ready
```

(8)

Interestingly, it is only the first call of the day that is negotiated so politely and verbosely. When other calls during the day occur they are either a direct call - without checking availability of the other - or some fast negotiation such as:

```
[15:28:16] IND: Hi DK...r u there?
[15:28:37] DK: yes
[15:28:53] ** Call to DK, duration 08:59. **
```

(9)

2) *The Collaboration Dimension*

Collaboration chats are technical discussions used not only whenever developers have doubts on how to develop something or on what an incident means, but also when they need clarification on what the software should do. Danish managers start collaboration chats with developers whenever they want to understand what a specific piece of software does or when they need to know the implementation choices of the developers.

```
[13:35:51] DK: Does your solution for 699 hide any field that is of type GUID? The main reason this came up is because when you do an Intersect or Union, the GUID from the two input
```

```
           feature classes are added to the output table. This is what DK wanted hidden, not just the standard "ID" field.
[13:36:21] IND: oh okay, I didnt do that :(
[13:36:49] DK4: Yes, I think the criteria should be if the field type is "Guid", not what the field name is.
[13:36:49] IND: but cant some other guid fields could be important.
[13:37:07] DK: Possibly, but for now we want them all hidden.
[13:37:23] IND: okay will do that.
```

(10)

Collaboration chats are less frequent than coordination chats, but they are definitely precious discussions in which design decisions are taken. For this reason these chats could be stored in a persistent channel. As reported by one practitioner:

> *In theory, you should write down in a design document all the things that you discuss. We do that, but we don't do for Skype conversations. We could attach the log to an incident.*

Collaboration chats complement formal design specifications; even if requirements are not very well specified, coordination chats help clarifying what an incident means, how to develop a specific piece of software and what the software should do under specific conditions. As observed, usually collaboration chats do not last more than 10 minutes, and thus technical issues are solved with very fast interactions between co-workers, probably faster than detailing formal design documents. Interestingly, managers agree on the fact that sometimes it is easier to have a direct discussion about a specification rather than detailing it formally. On the other hand, they do state:

> *Time wasted is somewhere else: if we don't have a proper design discussion, maybe something goes wrong and we have to do the work again later.*

So it would probably be necessary at times to dedicate more attention to these specific chats or to go into more depth with a voice call.

3) *The Awareness Dimension*

The awareness dimension is often part of the coordination chats. While people coordinate on their work, they also communicate with team members that they are working on some artifacts. Examples of a coordination chat with a strong awareness dimension are the coordination chats (3) and (4). Here, the coordination of the cooperation serves also to inform one the completition of a task the commencement of the next one. Whenever implicit mechanisms of awareness are insufficient, explicit *awareness chats* state what people are doing or when they will finish their work. In order to plan future work, managers ask developers what they are working on and request estimation of the time required to complete the work. Here are two examples:

```
[14:17:47] DK: Hi IND, looks like you are getting close to finishing your current tasks?
[14:18:52] DK: hmmm, and 559 is really not ready for execution as far as I can see....
[14:19:53] IND: Hi DK, not exactly.... ya you r right.... i m finishing my tasks asap but IN704 that is on hold and i m changing it to In progress is  really hard to complete....
[14:20:16] IND: i mean its a huge defect... so it will take lot of time to complete i think
[…]
[14:21:28] DK: ok. What is your estimate?
[14:21:44] IND: i think it will take at least 2 more days to complete
[14:22:15] DK: that's ok, any technical issues, or is it just hard work?
[14:24:26] IND: just lot of work to do for that with one technical issue […]
[14:24:37] IND: its just like 2-3 pages specifications
[14:25:03] DK: yes, just saw that. Happy coding!
[14:25:43] IND: thanks...
[14:25:44] IND: :)
```

(11)

```
[05:21:45] DK: IND, are you looking at the unit tests by chance?
[05:23:34] IND: yes
[05:23:37] IND: :)
[05:23:44] DK: ok, carry on :)
```

(12)

4) *The socialization dimension*

Socialization chats are useful for team building in increasing satisfaction and motivation of team members. We report three examples. Example 13 is an introductory socializing talk for establishing the collaboration that follows.

```
[11:44:22] IND: Hi DK
[11:44:44] DK: hi IND
[11:45:01] IND: hows monday treating you?
[11:45:16] DK: better than usual. Working from home
[11:45:25] IND: oh we r on the same boat
[11:45:41] DK: good to hear :)
```

(13)

Example 14 shows the good relationship that pairs have in the collaboration. After the technical discussion, the manager gives encouraging feedback to the developer and show that he appreciates the work that has been done.

```
[14:24:35] DK: uh - you are very methodic. That is great
[14:24:44] IND: Thanks
[14:24:55] IND: but if i have missed any..then please let me know
[14:26:31] DK: I will - but please home now :)
```

(14)

In Example 15 concerns a traffic enquiry, but the ulterior motive here is to understand what the working conditions of developers are during rush times in Delhi.

[08:43:05] DK: How is traffic?
[08:48:50] IND: traffic is mor organised these days
[08:49:17] IND: lot of security and traffic police man are deployed on the road...
[08:49:33] DK: ok, is it ok to get back and forth?
[08:50:05] IND: yep..for me esp...
[08:50:30] IND: bit problem near CWG village
[08:50:49] IND: localities but ok for me from gurgaon
[08:51:01] DK: ok, good to hear
(15)

As mentioned in the introduction to this section, the social dimension often exists only as a sub-text. An example provides chat interaction (4) where the developer wants to be sure about what he has to do next. This chat has a socialization sub-text. The development team has the tendency to communicate with managers in order to double check on what to do, rather than just checking on the formal tool. This kind of interaction helps to keep a good relationship between co-workers. Also the more formal negotiation of calls in the morning seems to serve as an extended 'Good Morning!' Discussing the researcher's suggestion of using tool-activated feedback to let people know of the status of a Spira™ entry, one of the project managers mentioned:

> *Our team is not so big, so I think it's important to take care about what is happening and give the right attention to each team member.*

I. Discussion

In this section, we discuss the characteristics of the usage of different communication channels used by the virtual teams. We highlight the central role that IM plays: the glue between the heterogeneous channels. Finally, we highlight the importance of real time chats from a team building perspective not only from a coordination perspective.

A. Ecology of channels

The many communication channels exist not only side by side; rather they form an ecology of channels, a socio technical communication system where different channels are used in a complementary way. The team uses a whole range of channels. Some are structured: the source code repository and Spira™ system provide possibilities to change the state of the common artifact and provide limited possibilities to communicate through annotations related to the source code units and tasks. Other channels are unstructured, such as voice call, IM and mail. These channels can be used more freely for multiple purposes. Communication in structured channels is often formal, which means that it is defined and constrained by rules of how the work has to be performed; in unstructured channels, both formal and informal communication can take place, e.g., mails can be used for formal notifications but also for jokes or social chatting. Such a system has thus far been described with respect to an individual's work practice [34]; below, we see how a team combines different channels.

In the CSCW discourse, it has long been known that structured channels alone are insufficient. They need to be complemented by the possibility for unstructured communication in order to negotiate how to use the structured channel [23]. The literature on global software engineering emphasizes the necessity of informal communication on unstructured channels in order to build trust and allow for informal participation [15]. In the recent research and tool support, one can observe a growing interest in the convergence of structure and structured channels; for example, the IBM Rational Team Concert [5] integrates a single application IM with structured channels, such as bug tracking, requirements definition, and versioning systems. Although technical integration is important – and our analysis indicates that the more adequate screen sharing possibility is not used as it is less integrated with the central communication environment than the Skype™ screen sharing possibility – the technology is only one side of the system we see at play.

The analysis shows that the project team has developed a common practice based on a set of social protocols [25] of how to use the different channels: Recurring patterns of coordination can be observed in our empirical material. A social protocol can be explicitly defined or it can develop implicitly with the evolution of the project. For example we observed that if a project member is present on Skype™, he is expected to answer a Skype™ request within a certain time frame. An explicit rule can be seen in Figure 1: in the DHI team, the reference number of Spira™ artifacts is used to relate to the same object through different communication channels.

Such social protocols [25] define for what purposes the different channels are used, how they are combined and how communication through different channels is related (incident number in our case). In [7], communication breakdown between sub-teams using the same set of tools is related to the use of different social protocols. Our analysis supports the importance of using the same tool and work not only with the same development environment when developing software in a distributed way, but also developing common social protocols through what sociology calls articulation work and meta-work [30]. Beyond this set of research, our analysis provides an indication that IM can play a special role in such sociotechnical communication systems.

B. IM as a glue between heterogeneous channels

Virtual teams combine different channels according to their communication needs. During the overlapping working hours, IM is used for negotiating the usage of other channels and raising activity on other channels to the awareness of team members, e.g., check in of source code or changing status of tasks in the Spira™ system. For chats used to negotiate phone calls, this phenomenon has already been investigated by Nardi [20] in 2000. In our virtual team, IM plays a crucial role not only as a way to negotiate calls, but to coordinate all other communication channels. Even when team members are talking and sharing their screens, they exchange lines of code, links or incident numbers through the IM. We see four characteristics of IM contributing to this glue function:

- As an *informal channel*, IM provides the possibility of talking about code, requirements, tests, and other artifacts at the same time. In IM chats, people clarify what is formally stated in other channels, complementing the structured information in a mail or in an issue tracker system. A hypothesis for future work would be that IM provides a possibility to mediate and facilitate the work with and on assemblies of work objects and boundary objects as described in [4] [19]. For dedicated and more complex discussions, the partners may decide to switch from text-based chat to a voice call or even e-mail.
- IM provides a *less obtrusive* way to initiate a contact than a phone or Skype™ call. It gives the accessed partner the possibility of finishing a small task before answering.
- IM provides *awareness information* both with respect to the availability of team members and in lieu of a notification system. Despite the fact that Spira™ keeps track of formal assignments, thus providing a mail-based notification system, this functionality is not used. Team members often notify each other of the assignment of a task through IM.
- The nearly synchronous communication through IM seems to also provide a means for *social communication* for building trust and social relationships with co-workers. This is further discussed in the next section.

C. The importance of the social dimension

Only a few of the Skype™ chats are purely social, as e.g., typical chats around the water cooler in co-located settings. This might be due to Skype™ being used at the physical desk where work takes place rather than in a place away from working tools and tasks. The social dimension plays a crucial role in the team, and chat supports this dimension. Almost every chat that we analyzed had a social dimension as subtext of the main topic of discussion. Through IM, PAMs encourage developers, give positive feedbacks on their work, and joke about each other's cultures. According to the team members, the possibility of announcing achievements (handing over tasks) and supporting one another through small remarks and "smileys" is one of the reasons of not using the notification functionality of the Spira™ system. In our empirical material, the social dimension is tightly interwoven with the actual implementation of the work, rather than being an addendum in addition to the work; this confirms what is stated by Hinds and Kiesler in [16].

Research on the social bonding in distributed work emphasizes the importance of seeing each other [11] and on simulating co-workers presence through advanced videoconferencing systems [18]. In our case, however, video does not seem crucial from a social perspective. The relationship between team members and the sociality of the team is kept more through personal or amusing chats rather than through a video link into each other's workplace. For team members, Skype™ seems to work as an effective substitute for face-to-face communication.

II. Conclusions

This paper has presented an ethnographic study on how different communication channels are used by a software development team distributed mainly between India and Denmark. We highlighted social protocols for the usage and the combination of different channels. Due to overlapping working hours, real time collaboration is possible across sites. IM can be seen as a type of glue between different communication and collaboration channels, and it contributes to socializing with co-workers. We identified four dimensions in chat logs: coordination, collaboration, awareness, and socialization. Although only one dimension is usually predominant, we observed that social dimension is often present as a sub-text. These social dimensions support the collaboration and help to keep a good relationship between team members.

This work is a qualitative study on only one small team, and thus considerations cannot be generalized for every software distributed team. Moreover, the analysis focuses on only the Danish/Indian side, and does not consider the manager located in Portland, USA. The collaboration with this manager was more challenging, both for Indians and for Danes, since they could not share any working hours. In fact, the manager in Portland was forced to use a different set of tools or to work during non-working hours in order to collaborate with the rest of the team. Chat in this case was not used as glue for collaboration; rather the more crucial role was played by emails.

There is a range of implications for future research. More empirical studies would help in understanding how the interlacing of different channels supports cross-site cooperation rather than focusing on individual tools. Other kinds of social software, such as microblogging, social networks and content sharing sites, could be explored. In our case IM seems to have become the space in which the collaboration around assemblies of work objects and boundary objects is mediated and facilitated. Re-analyzing the field material would contribute to a better understanding of how practitioners combine heterogeneous artefacts and communication channels to make GSD work. This, in turn, might be used to design better tools. This article only touches on the concepts of coordination mechanism and social protocols as a theoretical base for understanding the communication ecologies of distributed teams.

Acknowledgment

The authors would like to thank DHI Water and Environment for providing endorsement for this work. This research project became possible because of the collaboration of the WB project team. The authors wish to

express their gratitude to the team whose members were abundantly helpful and welcoming during the fieldwork.

References

[1] S.A. Bly and S.R. Harrison. Media spaces: bringing people together in a video, audio, and computing environment. Communications of the ACM, 36(1):28–46, 1993.

[2] E. Bradner and G. Mark. Why distance matters: effects on cooperation, persuasion and deception. In CSCW '02: Proceedings of the 2002 ACM conference on Computer supported cooperative work, 2002.

[3] B. Bruegge, A. H Dutoit, T. Wolf, Technische Universit, and D Garching. Sysiphus: Enabling informal collaboration in global software development Chair for Applied Software Engineering. In ICGSE '06 Proceedings of the IEEE international conference on Global Software Engineering, 2006.

[4] P.R. Carlile, A pragmatic view of knowledge and boundaries: Boundary objects in new product development. In Organization science, pages 442–455. JSTOR, 2002.

[5] L. T. Cheng, S. Hupfer, S. Ross, and J. Patterson. Jazzing up Eclipse with collaborative tools. In Proceedings of the 2003 OOPSLA workshop on eclipse technology eXchange, pp. 45–49. ACM, 2003.

[6] J. Connell, G. Mendelsohn, R. Robins, and J. Canny. Effects of communication medium on interpersonal perceptions. In GROUP '01: Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work, 2001.

[7] D. Damian, L. Izquierdo, J. Singer, and I. Kwan. Awareness in the wild: Why communication breakdowns occur. In Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on, pp. 81–90. IEEE, 2007.

[8] Y. Dittrich, K. Rönkkö, Jeanette Eriksson, Christina Hansson, and Olle Lindeberg. Cooperative method development. Empirical Software Engineering, 13(3):231–260, December 2007.

[9] M. Elliott and W. Scacchi. Free software developers as an occupational community: resolving conflicts and fostering collaboration. In GROUP '03: Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work, 2003.

[10] M G Farkas, Social software in libraries: building collaboration, communication and community online. Information Today, Inc., 2007.

[11] R. S. Fish, R. E. Kraut, R. W. Root, and R. E. Rice. Video as a technology for informal communication. Communications of the ACM, 36(1):48–61, 1993.

[12] C. Gutwin, R. Penner, and K. Schneider. Group awareness in distributed software development. In Proceedings of the 2004 ACM conference on Computer supported cooperative work, 2004.

[13] M. Handel and J. D. Herbsleb. What is chat doing in the workplace? In Proceedings of the 2002 ACM conference on Computer supported cooperative work, pp. 1–10. ACM, 2002.

[14] J. D. Herbsleb. Global software engineering: The future of socio-technical coordination. In 2007 Future of Software Engineering, pp. 188–198. IEEE Computer Society, 2007.

[15] J. D. Herbsleb, A. Mockus, T.A. Finholt, and R.E. Grinter. An empirical study of global software development: distance and speed. In Proceedings of the 23rd international conference on Software Engineering, pp. 81–90. IEEE Computer Society, 2001.

[16] P. Hinds, and S. Kiesler, Distributed work. The MIT Press, 2002.

[17] E. Isaacs, A. Walendowski, and D. Ranganthan. Hubbub: a sound-enhanced mobile instant messenger that supports awareness and opportunistic interactions. In CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves, 2002.

[18] P. Kauff and O. Schreer. An immersive 3D video-conferencing system using shared virtual team user environments. In Proceedings of the 4th international conference on Collaborative virtual environments, pp. 105–112. ACM, 2002.

[19] K. C. Kellogg, W. J. Orlikowski, and J. A. Yates, Life in the trading zone: Structuring coordination across boundaries in postbureaucratic organizations. In Organization Science, vol.17, 2006 p. 22.

[20] B. Nardi, S. Whittaker, and E. Bradner. Interaction and outeraction: instant messaging in action. In CSCW '00: Proceedings of the 2000 ACM conference on Computer supported cooperative work, 2000.

[21] T. Niinimaki and C. Lassenius. Experiences of Instant Messaging in Global Software Development Projects: A Multiple Case Study. In Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on DOI - 10.1109/ICGSE.2008.27, pp. 55–64, 2008.

[22] C. Ou, X. Zhong, R. Davison, and Y. Liang. Can instant messaging empower teams at work? In Research Challenges in Information Science (RCIS), 2010 Fourth International Conference on DOI - 10.1109/RCIS.2010.5507296, pp. 589–598, 2010.

[23] M. Robinson. Design for unanticipated use... In Proceedings of the 3rd conference on European Conference on Computer-Supported Cooperative Work, pp. 187–202. Kluwer Academic Publishers, 1993.

[24] C. Robson. Real world research: A resource for social scientists and practitioner-researchers. Wiley-Blackwell, 2002.

[25] K. Schmidt and C. Simone. Coordination mechanisms: Towards a conceptual foundation of CSCW systems design. In Computer Supported Cooperative Work (CSCW), vol. 5, pp. 155–200, 1996.

[26] J. Scholl, J. McCarthy, and R. Harr.A comparison of chat and audio in media rich environments. In CSCW '06: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work, 2006.

[27] Y. Segerstad and P. Ljungstrand. Instant messaging with Web-Who. International Journal of Human-Computer Studies, 56(1): 147– 171, 2002.

[28] L. Setlock, S. Fussell, and C. Neuwirth. Taking it out of context: collaborating within and across cultures in face-to-face settings and via instant messaging. In CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work, 2004.

[29] S. Smale and S. Greenberg. Broadcasting information via display names in instant messaging. In Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work, pages 89–98. ACM, 2005.

[30] A. Strauss. The articulation of project work: an Organisation Process. The Sociological Quaterly, vol. 29, no. 2, pp 163-178.

[31] M. Thissen, J. Page, M. Bharathi, and T. Austin. Communication tools for distributed software development teams. In SIGMIS CPR '07: Proceedings of the 2007 ACM SIGMIS CPR conference on Computer personnel research: The global information technology workforce, 2007.

[32] J. Thom-Santelli and D. Millen. Learning by seeing: photo viewing in the workplace. In CHI '09: Proceedings of the 27th international conference on Human factors in computing systems, 2009.

[33] C. Treude and M. A. Storey. How tagging helps bridge the gap between social and technical aspects in software development. In ICSE '09: Proceedings of the 2009 IEEE 31st International Conference on Software Engineering, 2009.

[34] T. Turner, P. Qvarfordt, J. Biehl, G. Golovchinsky, and M. Back. Exploring the workplace communication ecology. In CHI '10: Proceedings of the 28th international conference on Human factors in computing systems, 2010.

[35] D. Zhao and M. Rosson. How and why people Twitter: the role that micro-blogging plays in informal communication at work. InGROUP '09: Proceedings of the ACM 2009 international conference on Supporting group work, 2009.

Information flow within a dispersed agile team: a distributed cognition perspective

Helen Sharp[1], Rosalba Giuffrida[2] and Grigori Melnik[3]

[1] The Open University, Walton Hall, Milton Keynes, MK7 6AA, UK
[2] IT University of Copenhagen, Denmark
[3] Microsoft patterns & practices, USA
h.c.sharp@open.ac.uk, rogi@itu.dk, grigori.melnik@microsoft.com

Abstract. One of the hallmarks of a co-located agile team is the simple and open flow of information between its members. In a co-located setting, peripheral awareness, osmotic communication and simple information radiators support agile principles such as collective ownership, minimal documentation and simple design, and facilitate smooth collaboration. However in a dispersed agile team, where individual team members are distributed across several sites, these mechanisms are not available and information sharing has to be more explicit. Research into distributed software development has been tackling similar issues, but little work has been reported into dispersed agile teams. This paper reports on a field study of one successful partially dispersed agile team. Using a distributed cognition analysis which focuses on information propagation and transformation within the team we investigate how the team collaborates and compare our findings with those from co-located teams.

Keywords: Dispersed agile development, distributed cognition, qualitative study

1 Introduction

Global Software Development (GSD) is increasingly becoming normal practice in the software industry [1]. Organizations establish global software projects distributed all over the world, involving multiple teams located at different sites. It is claimed that GSD has advantages but it is also challenging because teams have to deal with temporal, geographical and socio-cultural distance, resulting in difficulties with division of work, inadequate communication, knowledge management, project and process management issues and infrastructure problems [2]. Several strategies have been suggested to overcome these challenges including reduced intensive collaboration [3], reduced temporal distance [3], increased formal documentation [2] and organizational factors such as processes, structure and goal alignment [4]. In contrast, agile methods depend on close collaborations, frequent informal face-to-face communication rather than lengthy documentation, and self-organising teams.

Despite these apparent differences, interest has been shown in assessing the viability of agile in GSD. Some studies suggest that agile practices mitigate GSD challenges, whilst others believe they emphasize the challenges [5, 6], yet others suggest that agile practices need to be modified for success.

Holmström et al. [7] specifically explored how agile practices can reduce three kinds of "distance" – temporal, geographical, and socio-cultural. They found specific agile practices to be useful for reducing communication, coordination, and control problems. Layman [8] suggests that methodologies dependent on informal communication can be used on GSD projects, despite geographic, technical, temporal and linguistic hurdles: an email listserv, globally-available project management tool, and an intermediary development manager who played a strong role in both groups.

Challenges of distributed agile build on those of GSD in general, and include lack of close proximity, lack of team cohesion, lack of shared context and knowledge and unavailability of team members [9]. Communication related issues are the major challenges when using distributed agile [5]. Several researchers claim that extending or modifying agile practices is necessary in GSD. Lee at al [10] reported that conventional agile methods must embrace more rigour and discipline in a distributed setting. Kirscher et al. [11] recommend Distributed eXtreme Programming (DXP), in which eight XP practices are seen as independent of the locality of the team and thus are practices that can be applied in GSD while four of them (planning game, pair programming, continuous integration, and on-site customers) are dependent on collocated team members and thus require alternative solutions to work in GSD.

As in GSD, much of the research so far into distributed agile teams focuses on the situation where multiple teams are distributed globally. To date, little research investigates dispersed teams [12] where most or all individual team members are alone, i.e. they are the only team member in any one location. This situation has become more relevant as experts are often widely distributed, and small open source projects [13] also have similarities to the dispersed model.

Previous analysis of co-located agile teams highlights that information flow within and around team members is simple and open, supported by few mediating artefacts that promote discussion [14]. In this paper, we use the same analysis approach to investigate information flow in a partially dispersed agile team and to compare the results. In the next section we describe the study situation including the team, the data gathering and analysis. The analysis itself is presented in Section 3, Section 4 discusses our findings in the context of co-located agile teams and the challenges of global agile development, and section 5 presents some conclusions and future work.

2 The Study

The project under study was to develop enterprise software components for use by software developers in their own organisations when building their cloud-based solutions. The deliverables are composed of binaries, tests and developer guidance.

The development, including 'spiking' iterations ran for 5 months from July 2011 to November 2011, although the initial product backlog had been developed over several months prior to this, through consultation with the user community and other development teams (this development is outside the scope of this study). The product backlog prioritization was heavily influenced by community votes.

2.1 The team

The project team consisted of one core team, an additional offshore testing team and a network of advisers (which was also globally dispersed). This study focused on the core team, which was made up of nine members. Most team members had worked together on a couple of previous projects, and hence knew each other. The team was an example of an agile dispersed development team as described above, although it should be properly defined as a partially dispersed team i.e. only one member of the team was in any one location for the majority of the time, except for those based in the team room in Seattle. At critical times within the product development (such as kick-off/exploration, beta release, final release) all team members who could attend, would visit Seattle for one or two sprints. The two development leads based in Buenos Aires would also meet occasionally and work together, although they mainly worked in different locations. Other team members would frequently pair remotely. Cross-discipline pairing (e.g. technical writer with developer) would also take place. An overview of the team members' roles and their locations is given in Table 1 (the real names of team members are substituted by pseudonyms to protect their identity). Only the documentation lead was a native English-speaker, but the main project language was English.

Table 1. Team membership

Pseudonym	Role	Location	Time zone
David	Technical writer, documentation lead	Bristol, UK	GMT +0:00
Edwin	Technical writer, training materials	Hague, The Netherlands	GMT +1:00
Mamu	Test lead (the majority of the testing team were located in India)	Vancouver, Canada	GMT -8:00
Rina	Tester	Seattle, USA	GMT -8:00
Eliah	Developer and subject matter expert	Seattle, USA	GMT -8:00
Joe	Lead developer	Buenos Aires, Argentina	GMT -3:00
Frederico	Lead developer	Buenos Aires, Argentina	GMT -3:00
George	Product owner	Seattle, USA	GMT -8:00
Jon	Developer (& user interface designer)	Montreal, Canada	GMT -5:00

The team used the hybrid "XP@Scrum" [15] approach with Scrum project management practices and XP engineering practices. Specifically, they met for stand-up meetings every day, developed in 2-week sprints with an iteration planning meeting, customer demos (with the product owner and the members of the advisory board) and a retrospective at the end of every sprint, had requirements expressed in stories, followed test-first development and practiced continuous integration and pairing. Within the two-week sprints, the team met every day for a 15 minute standup at 9.30am Seattle time, which fell within the working day for all timezones represented in the team. The furthest away was Edwin, in the Netherlands, for whom this time translated into 6.30pm. This meeting was accomplished using Microsoft's Lync, with some team members dialing in via computer telephony and others via regular phone lines. As typical to Scrum, each team member would report what they did before this meeting ("yesterday" for some team members and "today" for others, depending on timezone), what they are doing after this meeting ("today" or "tomorrow"), and any blocking issues. The offshore test team could not be present at stand-ups due to timezones so Rina acted as their proxy. Even when travelling, team members would make every effort to attend the standup. Once each team member had reported the team decided whether there was a need for a further discussion about any issues that had arisen (which was typically the case), and also when to do a triage. If it was decided that a discussion was required then the meeting resumed after a 10 minute break (for coffee). Sometimes all team members would attend these meetings and sometimes only relevant participants. Important meetings were recorded for future viewing by team members who could not attend e.g. the testers and technical writers. All team meetings were supported by Lync, with screen sharing being used for demos or to share diagrams etc. It has been reported by a team member that occasionally video streaming was used. Microsoft's OneNote and Team Foundation Server supported knowledge sharing. OneNote is a collection of wiki pages which can be tailored to any situation according to needs. For this project a shared OneNote file with revision tracking and residing on a Skydrive was used, so that everybody could contribute. Team Foundation Server houses the source repository together with the product backlog and sprint backlogs with the current status of stories and bugs within the backlog. Skydrive, a shared network folder for large files, audio recordings and presentations, was used by the team as an additional project artefact repository.

2.2 Data gathering

An ethnographically-informed approach was taken to data gathering [16]. One of the authors (the researcher) conducted an observational study of the agile team and one other author is a member of the team under study. The researcher observed all team members remotely, as a common practice in virtual ethnography [17], with the purpose of understanding team members' perspective of having virtual colleagues and virtual meetings. Additionally, following a strategically-situated approach [18], some sites have been visited in order to enhance the understanding of the team and to understand the role of physical artefacts in a partially dispersed agile team.

The researcher attended two or three stand-ups a week and eight iteration planning meetings over the project, joined three triage sessions and several ad hoc conversations, visited USA and UK sites and obtained photographs of the Argentinean environment. She also had access to the team's OneNote notebook which contained records of the team's retrospectives and many brainstorms and discussions. At the end of the project she was also given recordings of pairing sessions and other ad hoc meetings. The data collected included observation notes, screen captures of Lync conversations, still photographs, recordings of team conversations, pairing sessions and iteration planning meetings. In addition, as the issue of information sharing was a clear theme in the field study data, a short questionnaire was completed by each team member asking for details of how and when they shared information. The questions asked in this questionnaire are reproduced in Table 2.

Table 2. Questions asked of the team about information sharing

Information sharing questions
1. How do you share information with your fellow team members?
2. Apart from the daily stand-ups, how often do you contact your fellow team members, and how, e.g. telephone Joe every hour for 5 mins?
3. Are there artefacts or items of information that you don't need to share? Please describe specific examples and indicate why you don't need to share them.
4. If you come across something you don't understand, where do you go for help? Please provide three example sources and describe the kind of information you glean from each

2.3 Distributed Cognition

Distributed cognition theory [19] examines the cognitive processes that are dispersed among individuals and between individuals and artefacts in the external environment with and through which they interact [20]. Within CSCW and HCI it has been used to investigate collaborative working e.g. [21, 22]. This kind of analysis views a cognitive system as being distributed across individuals, artefacts, internal (i.e. cognitive) representations and external representations in the environment. It focuses particularly on how information is propagated and transformed within the system to achieve collaboration. Co-located agile teams have been analysed using this approach [14, 23], but its use in software development more widely has been limited (e.g. [24]). In this paper, we base our analysis on Distributed Cognition in order to investigate how information flows within, around and through a partially dispersed agile team, as compared with a co-located agile team. We draw on previous descriptions of distributed cognition, and a technique called DiCOT to analyse the team's information flows.

2.4 DiCOT (Distributed Cognition for Teamwork)

DiCOT [25] provides a structured approach to reasoning about a situation from a Distributed Cognition point of view. It draws on ideas and representations from Contextual Design [26], together with a series of principles that are central to distributed cognition. There are three main themes in DiCoT:

1. The *physical theme* focuses on the physical environment within which the cognitive system operates, at whatever level of granularity is relevant, from the building or office layout to the positioning of items on a desk or noticeboard.
2. The *artefact theme* focuses on the detail of artefacts that are created and used to perform the activity under study.

3. The *information flow theme* focuses on what and how information flows through the cognitive system, the media which facilitate that flow and how the information is transformed in the process.

Furniss and Blandford [25] identify 22 principles from distributed cognition which can be loosely categorised according to these three themes (see Table 3). Each theme can be investigated using these principles, an associated model, and a tabular representation to capture the detail of activity within a theme. Although further work has been done to extend DiCOT to two other themes, these three were used in the original analysis and for comparability we focus only on these three.

Table 3. The principles of Distributed Cognition underlying DiCOT

Physical Layout

- Space and cognition: considers the use of space to support activity, e.g. laying out materials
- Perceptual: considers how spatial representations aid computation
- Naturalness: considers how closely the properties of the representation reflect those of that which it represents
- Subtle bodily supports: considers what if any bodily actions are used to support activity, e.g. pointing
- Situation awareness: considers how people are kept informed of what is going on, e.g. through what they can see, what they can hear and what is accessible to them.
- Horizon of observation: considers what an individual can see or hear (this influences situation awareness)
- Arrangement of equipment: considers how the physical arrangement of the environment affects access to information.

Artefacts

- Mediating artefacts: are used to perform the activity
- Creating scaffolding: considers how people use their environment to support their tasks, e.g. creating reminders of where they are in a task
- Representation-goal parity: considers how artefacts in the environment represent the relationship between the current state and goal state.
- Coordination of resources: considers the resources (e.g. plans, goals, history and so on) that are co-ordinated to aid action and cognition.

Information flow

- Information movement: considers the mechanisms (representations and physical realisation) used to move information around the cognitive system
- Information transformation: considers when, how and why information is transformed as it flows through the cognitive system
- Information hubs: are a central focus where information flows meet and decisions are made.
- Buffers: hold up information until it can be processed without causing disruption to ongoing activity.

- Communication bandwidth: considers the richness of a communication channel, e.g. face-to-face communication imparts more information than email
- Informal and formal communication: recognises that informal communication can be very important
- Behavioural trigger factors: cause activity to happen without an overall plan needing to be in place.

3 DiCOT analysis

3.1 Physical layout to support cognition

In terms of the office or working environment, each location was different, but no evidence was found of the working environment being used to support activity. Most collaborative activity took place in a virtual setting, e.g. in a Lync meeting or through instant messaging. Very little collaborative activity had a physical aspect to it except the layout used in the software support tools (which we explore through the artefact theme below). To illustrate the physical environment of the team members, we consider the team room and two other example workspaces below.

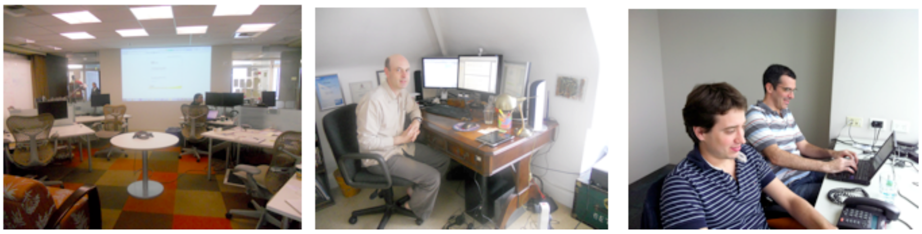


Fig. 1. (a) The team room in Seattle; **(**b) David's office; (c) Frederico and Joe while pairing

The team room in Seattle. Fig 1 (a) illustrates the team room environment. The walls are covered in white boards and several sketches and lists were on these walls at the time of the researcher's visit. However, only two walls were related to the current project, one of which contained a list of topics which had been identified for team conversations and another one was used for design discussions and sketching of specific features. The focus of the team room was twofold: the large screen which was used to display screen sharing during meetings; and the conference telephone on the small round table in the middle of the room. During meetings, team members present in the team room would sit or walk around this table. There was also a music centre for streaming music, and it was reported to the researcher that the team had an M&M's dispenser and adopted the ritual of getting a candy when a bug was fixed. In terms of the seven principles of the physical layout theme, the team did not in general use *space* or physical layout in the team room to support their work. However it was reported to the researcher that during onsite working, when more of the team members were present in Seattle, the physical space and layout were important. As there are no physical representations, the *perceptual* and *naturalness* principles are not relevant. During the researcher's time in the team room, there was no evidence of the use of *subtle bodily supports*, *situation awareness*[1] or *horizon of observation* to support collaborative working. This is not surprising as the team focus was elsewhere – as the researcher's notes comment, they were "somewhere in the ether". Arranging the large screen and the conference phone *equipment* in the centre of the room meant that everyone has clear access to information being shared digitally or orally, although each member of the team could also log onto Lync through their own computers to join the meeting.

Individuals' physical setting. David was based in the UK and worked mostly from home. The researcher visited his location to observe a day's work. The most striking aspect of David's environment was his computing *equipment*: an array of three screens sitting on his desk. While working, David would have several windows open spread across the three screens. There was no other evidence of the use of physical *layout* to support David's working, and as he was on his own no other physical principles are relevant here. Frederico and Joe came together to pair occasionally, and the only support they had in their physical environment was their laptops. Fig 1 (b, c) shows David's environment, and that of Frederico and Joe when they came together to pair.

3.2 Artefacts created or used

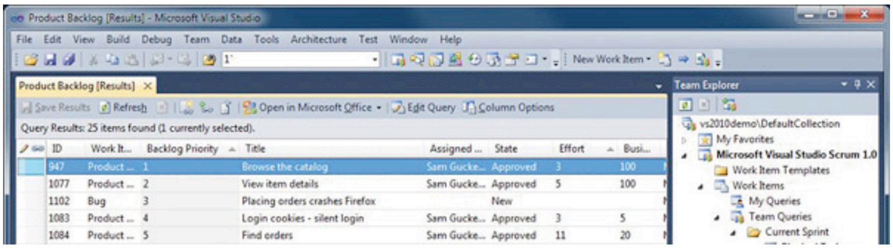
Among the locations observed, physical artefacts supported the team's work only in the office in Seattle – most of the distributed team's work and all activity of dispersed team members were supported through electronic documents and diagrams. The team room walls in Seattle (which were made of whiteboard material) displayed the list of topics for discussion, checklists and other information such as login details and configuration lines. There were also notes from brainstorming and design sessions. These artefacts were present only in the team room. The posters and certificates hanging on the walls in David's office, were not related to his work and the only sign of a physical external artefact was a (clean) pad of paper on his desk. In the office of Frederico and Joe, no physical artefacts were present at all, neither on the desk nor on the walls.

Regarding electronic artefacts, two main repositories supported the team's work: OneNote which the team referred to as their "Knowledge Base", and Microsoft Visual Studio Team Foundation Server (TFS). The latter is a collaboration platform to support teams through the automation and integration of processes, tools and project artefacts built around a central repository (with version control, build automation, workitem tracking etc.) as well as powerful reporting that help analyze and track progress and quality of the projects in real-time[2]. Our team used the facilities within

[1] However one team member reported that overhearing others' conversations did trigger decisions and other discussions

[2] http://www.microsoft.com/visualstudio/en-us/products/2010-editions/team-foundation-server/overview

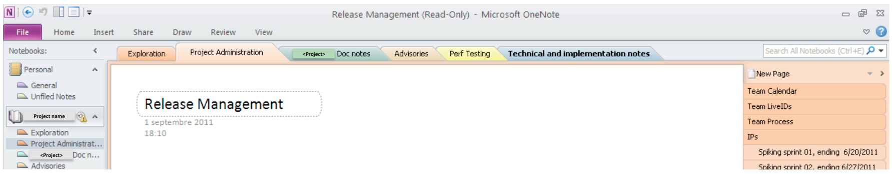
TFS to support an agile process, code sharing, bug tracking and to maintain the product backlog: the stories and their statuses (see Fig 3). OneNote is also a commercial tool, described as "a digital notebook" which supports the development and sharing of information using diagrams, text, schedules and so on. It also supports revision tracking, which is important for identifying updates made by collaborators. Several templates for OneNote are available, but custom structures can also be developed. The structure used by our team is illustrated in Fig 4.


Product Backlog [Results] - Microsoft Visual Studio
File Edit View Build Debug Team Data Tools Architecture Test Window Help
New Work Item
Product Backlog [Results]
Save Results Refresh Open in Microsoft Office Edit Query Column Options
Team Explorer
vs2010demo\DefaultCollection
My Favorites
Microsoft Visual Studio Scrum 1.0
Work Item Templates
Work Items
My Queries
Team Queries
Current Sprint

Query Results: 25 items found (1 currently selected).

ID	Work It...	Backlog Priority	Title	Assigned ...	State	Effort	Busi...
947	Product ...	1	Browse the catalog	Sam Gucke...	Approved	3	100
1077	Product ...	2	View item details	Sam Gucke...	Approved	5	100
1102	Bug	3	Placing orders crashes Firefox		New		
1083	Product ...	4	Login cookies - silent login	Sam Gucke...	Approved	3	5
1084	Product ...	5	Find orders	Sam Gucke...	Approved	11	20

Fig 3. Example screen shots of TFS (backlog view)



Release Management

1 septembre 2011
18:10

Fig 4. Example screen shot of OneNote. The tab structure reads: Exploration, Project Administration, <Project> Doc Notes, Advisories, Perf Testing, and Technical and implementation notes.

Other artefacts were shared through Skydrive, a shared network folder for large files, audio recordings, presentations. OneNote, TFS and files shared on Skydrive were used to co-ordinate action. OneNote included a team diary showing when individuals were on leave etc, together with contact details, and the product backlog aspect of TFS included showed who was responsible for which story and kept a record of who created or edited any documents. TFS was also used for bug tracking. A detailed discussion of OneNote, TFS and Skydrive is outside the scope of this current paper. In terms of the DiCOT principles, most of the *mediating artefacts* to support collaboration were electronic (as described above). When artefacts were considered interesting for the whole team and they were not digital (e.g. whiteboards sketches in Seattle office), photos of the whiteboard were taken and shared through Skydrive or OneNote for giving access to dispersed members. Generally, team members shared all artefacts with the rest of the team, but few exceptions occurred: David and other team members had private OneNote tabs where they kept their own notes separate from the shared set, which were used as *scaffolding* to support their own activity. Other note-taking and sketching behaviour that was observed included making personal 'to do' lists (e.g. by Eliah in the Seattle team) and drawing initial diagram sketches (e.g. by David before committing a diagram to a drawing package). The management of *representations* and *resources* focused on OneNote and TFS.

3.3 Information flow

Team members based in Seattle communicated through face-to-face conversations. The whole team communicated through email and Lync, mainly using one to one instant messaging chats, phone calls and screen sharing. Use of video and group text chat was very rare. Phone calls and screen sharing were only used when more detailed discussion was needed or for specific critical issues. Information was therefore *moved* around the team using each of these mechanisms, represented with dotted lines in Fig. 5. Team members were communicating together from few times per week to several times per day, depending on the role of each member and on the phase of the project.

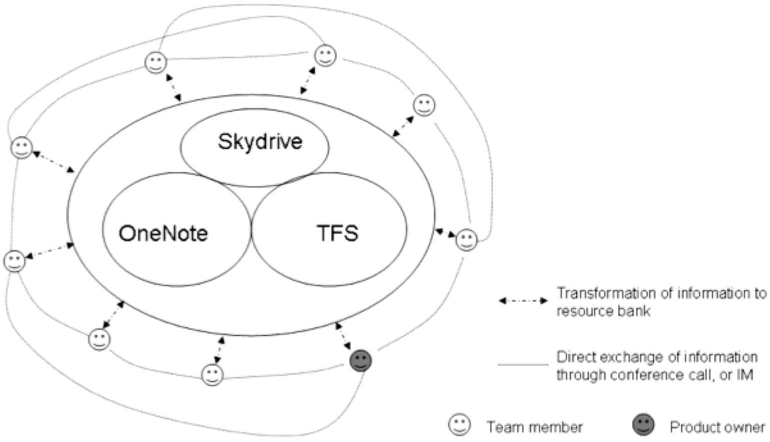

Skydrive
OneNote
TFS
Transformation of information to resource bank
Direct exchange of information through conference call, or IM
Team member
Product owner

Fig 5. Information flow within the partially dispersed team. OneNote, TFS and Skydrive were information hubs. Communication among team members occurred through face-to-face interactions, email and Lync. (note that all team members communicated freely with all others, and the communication lines are representative not exhaustive).

Since George was the product owner and Scrum master, he was regularly communicating with most of the team members as well as members of the other product groups and advisors and was managing the development of the project, therefore he was acting as an *information hub*. In addition, information was stored in OneNote, TFS and Skydrive and these were all significant *information hubs* and *information buffers*. Stand up meetings, triages and team conversations were also *information hubs* because key decisions were made at these times. The stand-up meeting was a co-ordination event but it was noticeable that during a stand-up, no documents were being shared. Individual team members took their turn to describe what they had been doing and what they will do next, as well as any blocking issues they had faced. During standups any additional discussion/brainstorming topics have been identified and scheduled promptly – often just after the standup. During the iteration planning meeting, screen and document sharing was more common, and for the sprint retrospectives an anonymized, shared note page was used for team members to write their thoughts and irritations anonymously before discussion; the page was later saved in OneNote. Team conversations, demonstrations, presentations, advisory board meetings and iteration planning meetings were recorded for later viewing by team members not present, or for re-viewing when documenting or testing stories. Recordings were stored through Skydrive and all retrospective comments were also stored in OneNote. These recordings were also *information buffers*.

When two team members are talking synchronously, then information is *transformed* the least. When information is entered into TFS, it has to conform to the specific fields and templates within the system. There was no evidence that this transformation caused any confusion or uncertainty, but nevertheless transformation was necessary. To capture information in OneNote also required some transformation – into a diagram or notes within the document structure.

Communication bandwidth varied from face-to-face groups meetings within the Seattle office or during the on-site meetings, to synchronous group conference calls, to recordings of conversations, instant messages, and one-to-one conversations. There were many different channels used for communication and there did not seem to be any concern or confusion over the type, frequency nor bandwidth of interactions.

Although there were regular team interactions, and a rhythm to the day and the sprint (as you would expect to find in any agile team), there was little communication that might be described as '*formal*'. Some demonstrations to the group of advisors were rehearsed and kept to strict time, and in that sense were 'formal' compared to the regular short interactions between the team members via Lync IM to ask for clarification or to ask for a synchronous conversation. However, the majority of the interactions were informal. Each member of the team knew when the stand-ups, iteration planning and other regular meetings were happening, and hence would be available through Lync on time for them. There was also an implicit agreement to block 2 hours after standup for team discussions. Apart from this, team members were self-organising and would attend to tasks and responsibilities as they arose. One factor which supported this way of working was that each person has their own and well-defined role (see Table 1). As such *behavioural trigger factors* were hard to spot.

4. Discussion

4.1 Agile Dispersed Development and Global Software Development issues

In the introduction we discussed potential benefits [1] and main challenges facing distributed [27] teams, and agile distributed teams [9]. The team described in this paper is dispersed for including in the team the most talented developers and subject matter experts beneficial for the project. The case we reported here is a partially dispersed team following the XP@Scrum distributed development approach and our findings are in line with studies stating that agile practices can successfully be adopted in GSD [7]. Despite the fact that team members were distributed across different time zones and were geographically dispersed, the team collaborated using agile practices in order to complete the project.

GSD highlights the lack of informal communication in distributed settings due to geographical distance and time-zone differences [27]. In our partially dispersed team there were some overlapping working hours among team members, so *synchronous communication* and collaboration was possible, mainly through the use of Lync for IM and phone calls. When this was not possible, team members adapted their working hours to those of remote colleagues (e.g. European members attending evening meetings) or they were recording the meetings to share them asynchronously. *Communication* was mainly *informal* and team members were easy accessible for impromptu conversations over Lync. Some team members reported that they were collaborating on a daily basis with some remote colleagues, even for several hours per day; no formal communication has been observed and documentation (e.g. shared digital artefacts, wiki pages, recording of the meetings) was limited to what is considered necessary for getting the work done, as in every agile project. This is in contrast with traditional GSD where detailed, comprehensive documentation as well as codified, explicit knowledge are considered necessary because communication is problematic and tacit knowledge is difficult to share [2]. *Pair programming* is a controversial practice that some authors consider very difficult to be performed in a dispersed settings [12] or even impossible [28] because pairs cannot sit side by side. In our team, pairing sessions were performed despite the geographical distance, using screen sharing and audio calls.

4.2 Co-located agile versus Dispersed agile

A previous distributed cognition analysis of a co-located agile team [14] identified three main observations:

1. There are few mediating artefacts in the system and those that do exist are simple and lack detailed information, which encourages discussion.
2. Information flows are simple and open, thus promoting situational awareness.
3. The team works in an information-rich environment. Information is both easily accessible and immediately relevant and applicable.

Comparing these points to the team in this study, there are some parallels but also significant differences. Our partially dispersed team relied on several digital mediating artefacts (OneNote, TFS, recordings etc). Each of these contained very detailed information, and the software tools (particularly TFS) had sophisticated structures which require more effort to learn to use. This is not to say that the team members showed any indication of difficulties, but information was less accessible to newcomers or outsiders than in the co-located situation. The detailed information available to the team through these artefacts led to an information-rich environment, but significantly more transformation between representations was needed.

The simple, open flow of information in co-located agile teams makes use of physical space and relies on face-to-face communication and on physical artefacts [23]. A central role is played by the Wall and the Story Cards, and situational awareness is high (see Fig 6). In the dispersed team, information flows were open because anyone could contact anyone else on the team, but they were also restricted because information flow needed to be explicit – there was no equivalent to peripheral awareness among dispersed members. Communication among dispersed team members occurred through *ad hoc* computer mediated interactions and it was necessary to explicitly store the information in *information hubs* in order to share it. Comparing Figs 5 and 6 shows that the study reported here highlights very different patterns of interaction: the information flow of our dispersed agile team is focused on OneNote, TFS and artefacts shared through Skydrive (see Fig 5).

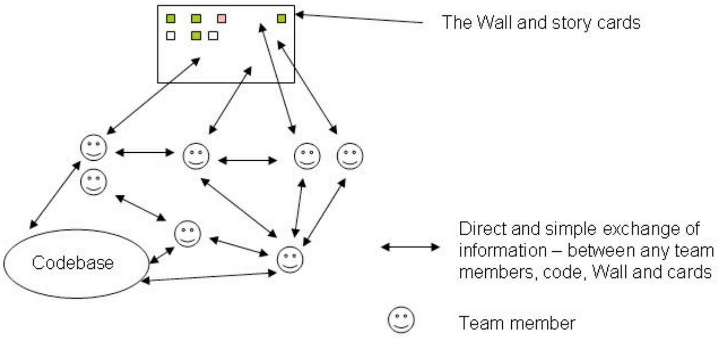

The Wall and story cards
Codebase
Direct and simple exchange of information – between any team members, code, Wall and cards
Team member

Fig 6. Information flow within a co-located agile team

During the meetings among dispersed team members there was no clear equivalent to the Story Cards and the Wall: screen sharing was used to focus attention of the team but often team members were not sharing screens, but just in audio contact. Moreover, the walls in the Seattle office were used for listing topics and checklists, or for brainstorming and design sessions; walls were not used as in co-located agile for organizing story cards or focusing attention during stand-ups. Since team members were dispersed, awareness of each other's activity was not as straightforward as in co-located settings and it was the responsibility of individuals to share information and artefacts with other team members. While in co-located agile an important role is played by the social context [29], in our partially dispersed agile team we observed a much stronger role for individuals deciding what to share and with whom.

4.3 Limitations

In this paper we reported only one case study of a small partially dispersed agile team. Although elapsed time covered the majority of development effort, the team has been observed for a limited period. Not all locations were visited and only limited on-site observation was possible – this limitation is mitigated by one of the authors being a member of the team.

5. Conclusions and Future Work

Our distributed cognition analysis of one partially dispersed agile team shows that the information flow within the team is more complex than that in a co-located team in the following ways: our dispersed team relied on complex digital artefacts with sophisticated structures rather than on simple physical artefacts used in a sophisticated way. It was therefore important for team members to be familiar with the tools being used. Information sharing needed to be explicitly accomplished, and information needed to be transformed more often than in a co-located setting. In addition, the responsibility of what information to share when and through which medium lay with individual team members. These are important points for anyone wishing to set up a dispersed agile team.

The team members themselves did not refer to communication as problematic. There were no references to communication problems in the records of the retrospective sessions, and although technology sometimes caused difficulties in the meetings, team members were not distracted or deflected by them, but simply continued with their activities. We did not investigate why this was the case, nor any other challenges and problems they faced. This may be the subject of future work. Other future work will include the study of further dispersed teams and a more detailed analysis of the team studied here. For example, Social Network Analysis (SNA) has been used to identify the relationships between distributed team members, e.g. collaboration patterns and impact of distance on awareness [30]. This kind of analysis could be used to investigate the role of artefacts and on how information is shared between dispersed team members.

Acknowledgments. We'd like to thank the team members for their support.

References

1. Conchúir, E. Ó., Ågerfalk, P. J., Olsson, H.H., Fitzgerald, B.: Global Software Development: Where are the benefits? Communications of the ACM. 52(8), 127-131 (2009)
2. Herbsleb, J.D., Moitra, D.: Global Software Development. IEEE Software. 18(2):16-20 (2001)
3. Carmel, E., Agarwal, R.: Tactical Approaches for Alleviating Distance in Global Software Development. IEEE Software. 18(2), 22 - 29 (2001)
4. Cataldo, M., Bass, M., Herbsleb, J.D., Bass, L.: On Coordination Mechanism in Global Software Development. In: Second IEEE International Conference on Global Software Engineering, pp. 71 – 80. IEEE, Munich (2007)
5. Hossain, E., Babar, M.A., Paik, H.: Using scrum in global software development: a systematic literature review. In: Fourth IEEE International Conference on Global Software Engineering, pp. 175–184. IEEE, Limerick (2009)
6. Jalali, S., Wohlin. C.: Agile practices in global software engineering-a systematic map. In: 5th IEEE International Conference on Global Software Engineering, pp. 45–54. IEEE, Princeton (2010)
7. Holmström, H., Fitzgerald, B., Ågerfalk, P.J., Conchúir, E.O´.: Agile practices reduce distance in global software development. Information Systems Management. 23(3), 7–18 (2006)
8. Layman, L., Williams, L., Damian, D., Bures, H.: Essential communication practices for extreme programming in a global software development team. Information and software technology. 48(9), 781–794 (2006)

9. Paasivaara, M., Durasiewicz, S., Lassenius, C.: Distributed Agile Development: Using Scrum in a Large Project. In: 3rd IEEE International Conference on Global Software Engineering, pp. 87-95. IEEE, Bangalore (2008)
10. Lee, G., DeLone, W., Espinosa, J.A.: Ambidextrous coping strategies in globally distributed software development projects. Communications of the ACM. 49(10), 35–40 (2006)
11. Kircher, M., Jain, P., Corsaro, A., Levine, D.: Distributed extreme programming. In: Extreme Programming and Flexible Processes in Software Engineering, Italy, (2001)
12. Braithwaite, K. and Joyce, T.: XP expanded: Distributed extreme programming. In: Baumeister, H., Marchesi, M., Holcombe, M. (eds.) 6th International Conference on Extreme programming and agile processes in software engineering. LNCS, vol. 3556, pp. 1524– 1526. Springer, Sheffield (2005)
13. Mockus, A., Fielding, R., Herbsleb, J.: Two case studies of open source software development: Apache and Mozilla. ACM Transactions on Software Engineering and Methodology. 11 (3), 309–346 (2002)
14. Sharp, H., Robinson, H., Segal, J., Furniss, D.: The Role of Story Cards and the Wall in XP teams: a distributed cognition perspective. In: Proceedings of the conference on Agile 2006, pp. 65-75. IEEE Computer Society Press, Minneapolis (2006)
15. Vriens, C.; Certifying for CMM Level 2 and IS09001 with XP@Scrum. In: Agile Development Conference, pp. 120-124. IEEE, Salt Lake City (2003)
16. Robinson, H., Segal, J., Sharp, H.: Ethnographically-informed Empirical Studies of Software Practice. Information and Software Technology. 49(6), 540-551 (2007)
17. Hine, C.: Virtual ethnography. Sage Publications Ltd, (2000)
18. Marcus, G.E.: Ethnography through thick and thin. Princeton University Press (1998)
19. Hutchins, E.: Cognition in the Wild. MIT Press, Cambridge (1995)
20. Hollan, J., Hutchins, E., Kirsch, D. Distributed Cognition: Toward a new foundation for human-computer interaction research. ACM Transactions on Computer-Human Interaction. 7(2), 174-196 (2000)
21. Furniss, D.: Codifying Distributed Cognition: A Case Study of Emergency Medical Dispatch. MSc Thesis. UCLIC (2004)
22. Halverson, C. A.: Activity theory and distributed cognition: Or what does CSCW need to DO with theories? Computer Supported Cooperative Work. 11, 243-267 (2002)
23. Sharp, H., Robinson, H.: Collaboration and Co-ordination in mature eXtreme Programming teams. International Journal of Human-Computer Studies. 66, 506-518 (2008)
24. Flor, N.V., Hutchins, E.L.: Analyzing distributed cognition in software teams: a case study of team programming during perfective maintenance. In: Fourth Workshop on Empirical studies of programmers. pp. 36-64. Ablex, Norwood (1991)
25. Blandford, A. Furniss, D.: DiCoT: a methodology for applying Distributed Cognition to the design of team working systems. In: Interactive Systems. Design, Specification, and Verification. LNCS, vol. 3941, pp. 26-38. Springer (2005)
26. Beyer, H., Holtzblatt, K.: Contextual Design: Defining Customer-Centered Systems. Morgan Kauffman, San Francisco (1998)
27. Herbsleb, J.D.: Global software engineering: The future of socio-technical coordination. In: 2007 Future of Software Engineering, pp. 188–198. IEEE, Minneapolis (2007)
28. Shrivastava, S.V., Date, H.: Distributed Agile Software Development: A Review. Journal of Computer Science and Engineering. 1 (1), 10-17 (2010)
29. Sharp, H., Robinson, H.M., Petre, M.: The Role of Physical Artefacts in Agile Software Development: two complementary perspectives. Interacting with Computers. 21 (1-2), 108-116 (2009)
30. Damian, D., Marczak, S., Kwan, I.: Collaboration Patterns and the Impact of Distance on Awareness in Requirements-Centred Social Networks. In:. 15th IEEE International Conference on Requirements Engineering, pp. 59-68. IEEE, New Delhi (2007)

How Social Software Supports Cooperative Practices in a Globally Distributed Software Project

Rosalba Giuffrida
IT University of Copenhagen
Software and System Section
Copenhagen, Denmark
rogi@itu.dk

Yvonne Dittrich
IT University of Copenhagen
Software and System Section
Copenhagen, Denmark
ydi@itu.dk

ABSTRACT

In Global Software Development (GSD), the lack of face-to-face communication is a major challenge and effective computer-mediated practices are necessary. This paper analyzes cooperative practices supported by Social Software (SoSo) in a GSD student project. The empirical results show that the role of SoSo is to support informal communication, enabling social talks and metawork, both necessary for establishing and for maintaining effective coordination mechanisms, thus successful cooperation.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management—*Programming teams*

General Terms

Human Factors

Keywords

Global Software Development, Communicative Genres, Coordination Mechanisms, Social Software, Socialization, Metawork, Articulation Work

1. INTRODUCTION

Software Engineering (SE) is a cooperative work [8]: software developers must coordinate their individual activities with tasks performed by other team members [20]. Coordination relies on communication — direct communication as well as communication mediated by code, documentation and artifacts. Communication is fundamental to coordinate the cooperative work and to establish and maintain effective collaboration. In Global Software Development (GSD) settings, effective coordination is challenging [12] due to the lack of face-to-face communication [3]. Research in GSD aims to overcome this challenge improving processes and tools for supporting cooperation among remote teams — see e.g. [12], [13], [9]. However, there is often a gap between SE processes and models suggested by literature and everyday practices of software teams.

Practices in situated action [23] often differ from plans [19]. Thus, adaptation by teams is necessary: SE methods, tools and processes need to be adopted by the team members and adapted to the team's necessities in order to establish shared practices. When team members agree on a set of rules, conventions and policies — the so-called *social protocols* [21] — the cooperative activity works smoothly. While face-to-face communication facilitates the negotiation of social protocols in co-located SE, establishing common practices can be seen as a major issue in GSD, due to temporal, geographical and socio-cultural distances [5]. However, GSD appears a good setting for studying and analyzing whether and how social protocols are negotiated and established: cooperative practices in GSD are mostly computer-mediated, thus traces of communication and coordination are mostly documented or recorded in the logs of the tools used by team members — e.g. issue management system, email, Instant Messaging chats, Wiki. In this setting, an important role is played by Social Software (SoSo) to foster, establish and keep social protocols within a GSD team; thus, studying the use of SoSo appears promising to understand the adaptation of processes and the adoption of tools in a GSD project.

This paper shows how SoSo supports remote cooperation complementing collaborative tools used in the everyday practices of GSD teams. Cooperative practices in a GSD project are analyzed and described through the analytic concepts of *coordination mechanisms* [21] and *communicative genres* [24], using a practice-based approach [16], to better understand what is necessary to establish and keep a satisfying cooperation. The paper shows that SoSo is a flexible channel which enables informal communication, supporting a wide range of communicative genres, such as metawork, social talks and work discussions. Socialization among team members through SoSo provides a context for the project, improving subsequent cooperation. Initial metawork through SoSo supports the effective establishment of shared coordination mechanisms and positive reactions to metawork proposals contribute to build good relationships among distributed team members.

2. RELATED WORK

GSD is increasingly becoming a common practice in the software industry [5]. Organizations establish global software projects, which are scattered all round the globe, involving multiple teams located at different sites. There are many

CHASE '14, June 2 – June 3, 2014, Hyderabad, India

potential benefits that can arise from GSD: it is promoted to lower development costs due to salary savings and to decrease development time due to time-zone effectiveness, to reduce time to market and to access the most talented developers [5]. However, GSD is a developing practice, thus there are challenges to overcome and problems to solve: since GSD is highly geographically dispersed, teams have to deal with temporal, geographical and socio-cultural distances [4], resulting in major difficulties in coordination and communication.

In distributed teams, informal communication cannot take place in person as easily as in co-located settings and the lack of face-to-face communication appears one of the main obstacles to cooperation [5]. However, research in GSD has established the importance of informal communication for cooperation in distributed settings — e.g. [9], [3], [14]. While, traditionally, main media for informal communication in distributed teams are email, phone and video conferencing system, nowadays informal communication take place also in the so-called Social Software (SoSo). Kaplan and Haenlein [15] define SoSo as "a group of Internet-based applications, built on the ideological and technological foundations of Web 2.0, that allow the creation and exchange of user-generated content". Essentially, SoSo encompasses a range of software systems that allow users to interact and share information, such as: Instant Messaging (IM), Internet Forums, Blogs, Microblogs, Wikis, Social Networks, Social Bookmarking. In a previous literature work [11], we collected research studies that emphasize the importance of SoSo to build and keep social relationships between distributed co-workers, to facilitate knowledge sharing inside the organization, to support knowledge management, to push informal communication between distributed team members. However, very few empirical works focus on the use of SoSo in GSD settings or analyze in detail how SoSo is used in relation with other channels adopted in the team [11].

In a previous study [7], we showed the flexible usage of SoSo, highlighting the central role of SoSo in the ecology of channels used by an established team. The paper shows that SoSo serves to support collaboration through other SE tools and that team members share conventions on how they use SoSo. This is in line with the fact that proper tool support is essential to overcome GSD challenges, but it is in itself not sufficient, as shown also in other research works. Damian et al. [6], for example, report a case in which differences in the usage of tools between USA and Canadian sub-teams cause a lack of awareness of changes to the code base. This breakdown occurred because the teams did not have a shared understanding of the usage of the tools; they did not share common *social protocols* [21] on how to coordinate about the changes. Since rules, e.g. in the form of methods and processes, under define practice, adaptation and appropriation of the methods are necessary. Another example is the work by Sigfridsson [22], that highlights the importance of metawork for the purposeful adaptation of practice. The analysis of everyday cooperative practices allows to gain insights on how SoSo is used to foster, establish and keep social protocols within a globally distributed software team, thus the role of SoSo appears interesting to understand adaptation of processes and adoption of tools in GSD.

3. THEORETICAL CONCEPTS

Since in GSD most of the activities are computer-mediated, the analysis of coordinative and communicative practices can be performed on the digital artifacts used and produced by the distributed teams during the remote cooperation. Thus, artifact-based concepts, such as *coordination mechanisms* [21] and *communicative genres* [17], appear appropriate to analyze and describe GSD cooperative practices. On the one hand, "a coordination mechanism consists of a coordinative protocol imprinted upon a distinct artifact which [...] stipulates and mediates the articulation of cooperative work so as to reduce the complexity of articulation work [...]" [21]. *Social protocols* [21] are a set of rules, conventions, policies shared by people involved in the cooperative activity. Social protocols develop practices and evolve as the project progresses; changes in the actual work can cause changes of the social protocols. On the other hand, a genre of organizational communication is characterized "by a socially recognized communicative purpose and a common form" [24]. The form of a genre refers to the "readily observable features of the communication, including structural features, communication medium and language" [17]. Coordination mechanisms and communicative genres will be used in the following to analyze and describe how coordinative and communicative practices are negotiated, established and maintained in a GSD student project and to understand the role of SoSo in the remote cooperation.

4. CASE DESCRIPTION

The project under study is part of a GSD student cluster in collaboration between IT University of Copenhagen and Peking University. The project was carried on by two remote sub-teams located in Denmark and in China, and took place from February 2011 to May 2011. An academic supervisor provided the description of the product to be developed and he evaluated the work performed by the students based on the code developed, on a final report produced by each sub-team and on an oral exam. The goal of the project was to design and implement an e-collaboration tool. The system design, the requirement specifications, the development of the product and the organization of the collaboration were students' responsibilities; therefore, the students were "self-organising" their work, sharing roles, responsibilities and decision taking. The Chinese sub-team was composed by three Chinese members: Cheng, that acted as a team leader of the Chinese sub-team, Wang and Wusheng. The Danish team was composed by five members coming from different European countries; the Danish team assigned responsibilities to each team member: Arnold acted as a Scrum master, Jakob was the main developer, Geaorgios was a developer responsible for the interface, Stella was responsible for the communication and for the final report, and Morten was a developer responsible for the database. Names are modified for privacy reasons. The team adopted Scrum as a development model and used it rather rigorously, e.g. organizing the project in five Sprints, producing backlogs and weekly standup meetings, which were locally performed and video recorded, then shared with remote team members. Each team started working on different ends of the system: Danish team was working on the front-end and on the database; Chinese team on the back-end; the teams then combined the work in the intermediate layer of the solution, without encountering major difficulties.

Tools used by the team are: Skype[1], for video conference

[1] http://www.skype.com/

and IM chats, a traditional phone, emails and Assembla[2], an issue tracker system with additional functionalities such as file sharing, Wiki, Forum and an integration to Subversion (SVN) [3], the version control system used. Assembla was used for keeping track of the status of the project, for managing issues, for defining deadlines and as a shared repository for file exchange. Most of the communication took place in Assembla, IM chats and during the weekly meeting. Email were rarely used, the forum of Assembla substituted the email for communication: it worked as a common repository of the messages exchanged by team members; moreover, it was possible to enable the notification via mail functionality that updated team members about the new events of Assembla. Despite time differences, IM chats were used by some team members, thanks to their flexible working hours and to the semi-synchronous communication offered by the tool. A Skype group chat with all team members was carried on during the whole project; it was mainly used in the initial phase of the project, though. Being a student project, team members were not working on a daily basis on the product and no fixed working hours were imposed. However, the team defined two fixed days a week for working on the project; due to the time difference of seven hours, there were no official overlapping working hours between team members, the only exception for both teams was the one hour weekly meeting in which synchronous communication was possible: every Friday at 1 p.m Danish time, i.e. 8 p.m. Chinese time. Skype resulted in very bad quality for video conference, thus the meetings were carried on through a normal phone for the audio; however, a visual channel was kept through Skype with mute audio. Despite distance, team members managed to have a smooth collaboration, without major frustrations; the team succeeded in having a working prototype; in the final report and in the interviews team members reported positive impressions towards the collaboration; finally the researcher noted during on-site observations and in the analysis of the field material a positive evolution of the project and the establishment of a good working environment.

5. METHOD

This research is carried out through an ethnographically-inspired approach: the first author attended most of the meetings from the Danish site and visited the Chinese team for a week, collected pictures, took notes during observations and interviewed participants. The team provided the researchers with free access to the whole Assembla repository, including Forum discussions, documentation, Wiki, files exchanged and source code. Moreover, the researchers had access to the Skype logs, to some emails and to the final report produced by the Danish team. The analysis is mainly performed on the Assembla Forum, on the Wiki, on the group chat and on two one-to-one IM chats; nevertheless, data from mail, the final report, and other Assembla sources are used as well to support evidences provided by the main analysis and to triangulate the findings. The first author coded all conversations carried on in the Forum and in the Wiki pages of Assembla, identifying communicative genres. She incrementally developed the coding schema while coding the messages; the coding schema was then discussed with the second author. Moreover, in order to limit research biases, a colleague not involved in the project was asked to check the coding schema and to independently code a sample of the messages; divergences in the coding were discussed and resolved. Finally, the same coding schema has been used to analyze IM chats — both group chat and one-to-one chats. The coordination mechanisms have been identified by the researchers going through artifacts and documentation available in the Assembla repository. Findings were triangulated with first author's observations and with semi-structured interviews with team members, both in the Danish and in the Chinese site.

6. ANALYSIS

This section describes which coordination mechanisms the teams established to accomplish their work and it shows the evolution of the communication over time, as it happens in the main communication channels used. The analysis illustrates how cooperative practices evolved during the project and it is performed on a weekly base: being a student project, the work is concentrated in few days a week, thus a weekly perspective provides sufficient information about the cooperation.

6.1 Coordination Mechanisms

Six main coordination mechanisms (CM) have been identified and are described in the following.

CM1: File Sharing. In the file sharing system, important files (*artifacts*) were regularly exchanged and, when the file was considered of particular relevance, a message was posted in the Forum to notify to other team members that the file was available in the system (*social protocol*). The first files were exchanged in Week 1, both by Danish and by Chinese sub-teams. The file sharing system was used throughout the whole project by both sub-teams until the last week of the collaboration. Several kinds of files were exchanged, such as: diagrams, images for the product interface, zipped files with source code, documentation, minutes of the meetings. The latter deserves a categorization on its own.

CM2: Minutes Sharing. Producing and sharing the minutes of the meeting was an initiative of the Danish sub-team and it was not a practice commonly defined or requested; however, it became an established practice in Week 4. The minutes were produced by the Danish sub-team starting from the first week of the cooperation: however, in Week 4, the Danish sub-team started to share the minutes, including the ones previously produced, and the minutes sharing coordination mechanism' got established. Chinese members considered it very important to overcome the language and the technological barriers: the minutes helped them to go through the discussions occurred in the meeting and check whether their understanding was correct or not.

CM3: Issue Managing System. Initial issues were created in Week 3 and the issues managing system was used and updated during the whole project, until Week 12. 110 issues have been created and used for coordinating the software development activities. Issues (*artifacts*) were created by all team members in the Danish sub-team and only by the team leader of the Chinese sub-team, Cheng. Issues were assigned to all team members, that completed the issues or re-assigned them to other team members (*social protocol*).

[2] http://www.assembla.com/

[3] http://subversion.apache.org/

CM4: Subversion (SVN) Repository. In Week 3, the team started to use SVN, the version control system for sharing the source code (*artifact*). Commits in SVN were regularly supported by comments about changes implemented (*social protocol*); SVN was used until the end of the project.

CM5: Standup Meeting. In Week 4, the first standup meeting was video recorded (*artifact*) by the Danish sub-team. Chinese team started to record and share standup meetings in Week 5. Video were uploaded in a private web server and the associated link was shared through a Forum message in Assembla. The video recorded standup meeting consists in about 5 minutes video in which each team member provides a status update to other team members. One standup meeting was independently recorded and shared by both sub-teams every week (*social protocol*). Both sub-teams reported that this practice goes beyond pure coordination: the advantage is that team members can clearly see each others, getting familiar with the environment where the remote team works, getting used to the accents of different participants, thanks to the possibility to watch the video several time, if necessary.

CM6: Agenda. The Danish sub-team started to share an agenda for the meetings in Week 1; the agenda was shared during the meeting through the Group Chat. In Week 4 a problem was reported in a Forum message by the Chinese sub-team in reaction to a not successful meeting: the Chinese sub-team explicitly requested to share the agenda prior to a meeting:

> We, PKU, advise that you, ITU, tell us agenda in advance. In that case, we, PKU, have time to prepare for it. And our meeting will be better. O(∩_∩)O *[Wusheng, China]*

In the same week, the Danish sub-team produced a document about the collaboration, in which rules for the sharing of the agenda were proposed:

> A written meeting agenda should be uploaded at Wednesdays in order to accomplish effective virtual meetings. The author of these should be shifting between China and Denmark. So e.g. 1: one team post a agenda proposal on Wednesday 2: the other team confirm, write a respond to it or add things to the agenda latest 2 hours before the meeting on friday. 3: the team who originally created the draft will add a final agenda latest 1 hour before the meeting starts.

The suggested rules were discussed in a Forum thread in Week 5 and the coordination mechanism got adopted and established in the same week: in Week 5 the agenda (*artifact*) was shared by the Chinese sub-team through a Forum message, it was integrated by the Dansih sub-team and finally modified by the Chinese sub-team (*social protocol*). This practice was carried on alternatively by the two sub-teams similarly during all the remaining weeks.

6.2 Communicative Genres

Communication among team members occurred mainly through SoSo and during the weekly meetings. Communicative genre analysis has been performed in all logs of the Group Chat, Wiki, Forum and in two one-to-one IM chats occurred between the Chinese team leader, Cheng, and two members of the Danish team, Stella and Jakob. Other one-to-one chats were not provided to the researchers, thus, it was not possible to include them in the analysis. Table 1 shows the amount of instances of communicative genres coded in relation with the different kinds of SoSo (*form*) and the various *purposes* identified[4]. Most of the communication happened in the Forum and in the Group Chat; Wiki pages were rarely used; Cheng did not have many conversations with Stella, while a relevant amount of discussions were carried on in the one-to-one IM chat with Jakob. This section provides an analysis of the evolution of the communication over time, relating it with the coordination mechanisms established in the team. Figure 1 synthesizes the analysis.

Work. Work discussions can be related to different topics such as decision making, requirement specification, technical issues, etc. This genre usually starts with a question, followed by one or more answers, that sometimes lead to a broader discussion. The Work genre is identified in all kinds of SoSo, but WIki. In particular, work discussions occurred mainly in the initial phase of the project until the coordination mechanism of the standup meeting (CM5) was established in Week 5. Nevertheless, work discussions start again at the end of the project when teams needed to integrate the different pieces of code. In this phase, few instances of Work genre appear in the Forum and in the Group Chat, however, an important role is played by semi-synchronous communication through one-to-one IM chat between the Chinese team leader, Cheng, and the main Danish developer, Jakob.

Knowledge Sharing. Genres identified with the purpose of knowledge sharing can be tutorials, how-to, solutions to potential problems. It is usually a message written by one team member to share his knowledge with other members. It includes a technical description of a problem and a suggestion on how to solve it. Often it provides links to other tutorials available online or to other sources, such as a tool webpage or documentation about API. Most of the Knowledge Sharing chats appear in the initial phase of the project in the Group Chat and during the project in some of the one-to-one IM chats. However, the Wiki and the Forum function as a permanent support for reporting some of the knowledge in a more structured and persistent way.

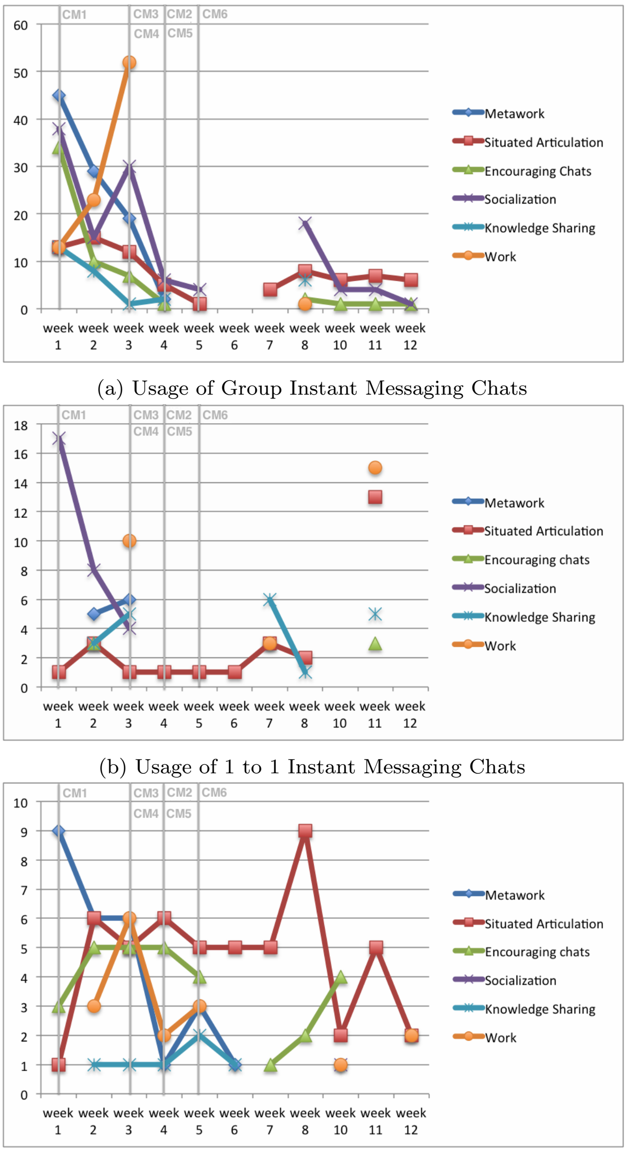
Metawork. Metawork communication consists in meta-comments about how to structure the work within the team. It can be proposals, questions, answers or discussions about how to carry on the work. Most of the metawork occurred in the initial phase of the project, especially through the Group Chat. However, also in the Forum, metawork discussions occurred in order to suggest how to establish coordination mechanisms. An example of metawork Forum message is reported below:

> Hello, everyone of ITU, I made a table by Word. In order to find the best time for our meeting,

[4] Instances are considered as Wiki pages, Forum messages and chunks of IM chats. The same instance can have more than one code. In chats, an instance is the minimum meaningful chunck of information exchanged, e.g. when one person writes one sentence in several lines of chat, the whole sentence constitutes the chunck of information.

Table 1: Number of instances of communicative genres identified

Purpose Form	Metawork	Situated Articulation	Encouraging Chats	Socialization	Knowledge Sharing	Work	**Total**
Group Chat	95	77	57	120	30	89	468
IM Cheng-Jakob	11	26	6	8	19	28	98
IM Cheng-Stella				21	1		22
Forum	21	51	29	1	3	17	122
Wiki	6				3		9



(a) Usage of Group Instant Messaging Chats

(b) Usage of 1 to 1 Instant Messaging Chats

(c) Usage of Forum and Wiki

Figure 1: *Temporal overview.* The amount of instances of communicative genres identified in the SoSo communication among team members. Vertical lines show when the six coordination mechanisms are established. Week 9 is not reported in the Figures because it was an holiday week, thus no communication happened among team members during that week.

> please download the .doc file and fill the table, and then upload it in a reply message. By it, we can have a clearly view which is the proper time. Thanks! *[Cheng, China]*

Thanks to metawork, effective negotiation of social protocols and decisions about coordination mechanisms have been taken and metawork discussions drastically drop in Week 4, both in the Forum and in the Group Chat, when some coordination mechanisms started to get established. However, the re-discussion about the use of the agenda in Week 5 brought further metawork in the Forum that disappeared once the coordination mechanism of the agenda (CM5) got discussed and established.

Situated Articulation. Situated Articulation communication are question, answers or informative messages about the work that has been done so far, in order to provide awareness to all team members about the status of the project. The Situated Articulation genre is present in all kinds of SoSo, but Wiki. In the one-to-one IM chats, the Situated Articulation is mainly related to awareness, in which team members ask for updates or provide information about the status of the project; in the Forum, the Situated Articulation genre is particularly interesting because it is used for supporting established coordination mechanisms. In particular, a large amount of "File Notification messages" is reported: these messages consist in notifications about the sharing of files, of minutes, of the agenda and of the standup meeting that supported the associated coordination mechanisms. It is interesting to note that, once coordination mechanisms got established from Week 5, patterns of collaboration are clearly visible in the Forum, e.g. as sequences of messages: standup, agenda, minutes.

Social Talks. Socialization chats consist in messages with the main purpose of socializing with remote team members, talking, for example, about each others' country; an example of IM one-to-one chat is reported below:

> *Cheng:* Denmark is a beautiful country. I will go to Denmark to have a tour if there any chance, To got to know this country
> *Stella:* well please come... It is beautiful!! I have several chinese colleagues and friends at work (sushi restaurant)

It is interesting to note that most of the one-to-one chats occurring between Cheng and Stella have a socialization purpose. Socialization chats appear only in the Instant Messaging, both in the one-to-one chats and in the Group Chat. In particular, they happen mostly at the beginning of the collaboration. Some socialization chats are intended for gaining contextual information about the project, as in the following example:

> *Cheng:* how busy are you with courses? Do you have time to work on the project?

Once the standup meeting got established as a coordination mechanism (CM5), socialization chats dropped, as social chats were happening through the standup meetings themselves and during the weekly meetings. The socialization chats appear again in the Group Chat in the final phase and they are jokes and funny chats that show the good relationship that has been established among team members:, while discussing work-related topics, students have fun about each others' culture and habits, e.g. exchange of messages in Chinese and lots of smiles. Social chats were mainly pushed by Cheng, Stella and Jakob, that can be considered the members that mostly tried to create a bridge among the two sub-teams, acting as *cultural liaisons* [3] for the team.

Encouraging Chats. Encouraging messages are positive feedback to a message posted by another team member, that can belong to any of the previous genres. An example is provided below:

> You did a great job! 佩服你们。 And we will finish our part of self-introduction soon.
> *[Cheng, China]*

Encouraging Chats are present during the whole collaboration, mainly in the Forum and in the Group Chat. Positive feedback on the work of the remote colleagues and appreciation on their work or metawork proposals are frequent in both sub-teams and seem related with encouraging good collaboration, providing a positive attitude among team members. They are rarely present in the one-to-one IM chats, except than in the final phase, when closer cooperation occurred in this medium between Cheng and Jakob. Encouraging chats appear in Week 7, 8 and 9 in the Forum, as a positive response to the funny standup meetings shared.

7. DISCUSSION

The previous Section illustrates communicative and coordinative practices of a successful GSD student team. This Section investigates the possible reasons for the success of the cooperation and discusses the findings of this paper.

7.1 Challenges Encountered

The main challenges the team encountered during the collaboration were: (a) language barriers: no one was native English speaker, Chinese were less self-confident with English than Danish team members; (b) technological barriers due to low bandwidth connection that causes troubles in the weekly video conference meetings; (c) different professional background: Chinese sub-team did not have knowledge or experience with Scrum methodologies.

The main challenge the team experienced was the language issue during weekly meetings due to strong accent on both sides; moreover, the low bandwidth connection with China aggravated the language difficulties and misunderstandings, causing breakdowns and frustrations among team members, especially in the initial phase of the project. However, team members tried to find a workaround for barriers in the cooperation. As an example, the video recorded standup meetings helped to overcome low bandwidth connection and language barriers, enabling an easy way to asynchronously exchange information with remote team and providing occasions for informal jokes and fun. Also the combined use of Skype for video and of the phone for the audio helped the team to carry on the weekly meetings avoiding frustrations caused by technological barriers. Professional inexperience of the Chinese team with Scrum methodologies was compensated through the usage of SoSo for knowledge sharing: in the group chat and in Wiki, tutorials and documentations were provided to share the knowledge about e.g. the methodology used. Good relationships among team members, established through social chats, enabled them to find a workaround for barriers in the cooperation; moreover, the high level of metawork in the initial phase of the project allowed to establish effective coordination mechanisms; finally, SoSo turned to be a flexible channel that allowed to overcome the lack of face-to-face communication and to complement collaborative SE tools, supporting the establishment and mantaining of coordination mechanisms. These aspects will be detailed in the following paragraphs.

7.2 Social Talks Enable Good Cooperation

Social talks largely appear in the initial communication through IM, both in the one-to-one and the in the Group Chat. This channel allows team members to get to know each other and to establish informal relationships, which allowed a positive attitude for further negotiations, resolution of conflicts and establishment of effective cooperative practices. Moreover, the social chats helped in providing a context for the cooperation, offering background information on the project and on the team members. Socialization appear fundamental to enable good work, in line with previous research — e.g. [18]. Through the analysis based on communicative genres and coordination mechanisms, it is possible to show how and why good practices come about. Moreover, in GSD research, it is often suggested to physically visit the remote site or to foster team building through ad hoc activities in order to reach informal relationship among distributed team members [3]. The team studied in this work overcomes challenges such as the lack of face-to-face communication or the low bandwidth connection mainly through the establishment of effective practices supported by SoSo. Nevertheless, an important role in helping the two sub-teams to overcome distance and to foster a sense of cohesion seems to be played by some team members, in literature often referred as *cultural liaisons* [3]: social chats are indeed pushed mainly by Cheng, Stella and Jakob. Especially thanks to these members, the team managed to develop a *negotiated project culture* [2] that permitted team members to effectively work together, despite being distributed and belonging to different cultures. Social talks supported by SoSo and pushed by the three team members enabled good collaboration and helped to find a workaround for the barriers in the cooperation. The personal attitude of these team members in fostering social talks appears influential as it lowered the potential socio-cultural distance among teams.

7.3 Metawork for Establishing Effective Coordination Mechanisms

In the initial phase of the project, thanks to the social talks, the two remote sub-teams established an effective informal channel, where metawork discussions could take place. The importance of metawork has been highlighted in previous research [22]; however, this study shows how metawork takes place and how it evolves, especially for establishing effective coordination mechanisms. The high level of metawork in the initial phase of the project, both in the Group Chat and in the Forum, shows an effort by the two sub-teams to discuss how the collaboration should be carried out; the metawork chats go along with encouraging chats and smileys, indicating acceptance of the proposals put forward by the remote team members. Metawork disappears in Week 5 when all the coordination mechanisms got established and team members start to work effectively thanks to commonly adopted coordination mechanisms. The team succeeded in establishing effective coordination mechanisms and social protocols thanks to the initial metawork and the associated encouraging and social chats, supported by SoSo.

7.4 The Role of Social Software

The analysis performed in this paper highlights the central role of SoSo as an informal and flexible channel that supports different kinds of communicative genres, such as work discussions, metawork, situated articulation and social chats. This is in line with previous research that highlights the possible usages of SoSo [11], though the six communicative genres identified in this paper have not been previously explored. SoSo has been researched and it is widely used in OSS [10], however the analysis presented in this paper helps to show the complementing role of SoSo in respect with other collaborative SE tools that provide templates for coordination mechanisms. In particular, it is interesting to note the support of SoSo as an informal channel in the initial phase of the project, when decisions need to be taken, social relationships need to be established and social protocols need to be negotiated. After the initial phase, Forum and Wiki serve as a persistent repository for the knowledge shared, while communication through Forum and IM allows situated articulation, decision making and collaboration record.

A special role is played by IM chats and it deserves further discussions. At the beginning of the cooperation, team members need to define and establish common social protocols and IM is fundamental in this phase, being used for socialization and metawork. Figure 1a and 1b show that in Week 5, 6, 7, and 8 there is not much communication in the chats, because coordination mechanisms are established and are supported by situated articulation through Forum Messages (Figure 1c). The collaboration through IM chats does not exist when team members have another working channel and effective coordination mechanisms are in place. At the end of the project, when more synchronous collaboration is needed in order to merge the two pieces of code developed, communication through chat appear again. This can be explained with the *glue concept* we introduced for IM in a previous work [7]: when the team is established, as in [7], chat is a glue between other channels and works as a situated dispatcher: through chat team members coordinate meetings, propose to move to other channels, e.g. mail, calls [7]. When closer collaboration is required, as in some examples reported in [7] and in the initial and in the final phase of this project, IM works as a realtime communication channel among team members. Finally, socialization happens mostly in the IM chats as a subtext [7] or as a genre on its own, as in the case illustrated in this paper.

8. LIMITATIONS AND FUTURE WORK

The main limitation of the present study is that it is based on a single student project. Being a qualitative study, generalization can be done by triangulating the findings with other studies. For example, results of this paper are in line with the work of Sigfridsson [22], that highlights the importance of metawork; in addition, the present article shows how metawork is supported by SoSo and how it is developed in the team, thanks to the communicative genre analysis. Though our case is a student project, Sigfridsson work suggests that similar processes take place also in professional contexts. Moreover, in this work, the project setup is not artificially imposed by any experiment-like constraint and it shows how a software product is developed, from the design phase until the delivery phase, in a distributed team, dealing with geographical, temporal and socio-cultural barriers [5] that characterize real-world GSD projects. Though the product developed is relatively small — it required only three months part time development — students were asked to act as professionals, dealing with GSD challenges on their own. Practices observed in this student project provide insights about how novices seek channels to communicate in software development. The commitment and the responsibility of the students to accomplish their tasks were not dissimilar to professional developers. Some of the students were part-time software developers in industrial settings. Undoubtedly, additional cooperative practices can be found in other cases: supplementary coordination mechanisms and communicative genres can be identified, other kinds of SoSo could be adopted by team members, other techniques can be used to overcome the lack of face-to-face communication. However, the focus of the research presented here is not on the quality of the code produced nor on the evaluation of specific methods and tools, but on how the team establishes the cooperation across distance, allowing to study how practices are negotiated, established and maintained in a GSD project.

Three lines of future research arise from this work. Firstly, the concept of culture has been briefly touched in the discussions of this paper, as it did not appear fundamentally affecting the cooperation. However, to reliably affirm that, comparisons with other teams with similar background should be performed. Moreover, culture should be analyzed with appropriate analytic concepts in order to establish whether and how the cultural liaison role [3] of some team members affected the cooperation.

Secondly, we can certainly affirm that social chats appear fundamental to support trusting relationships, thus enabling good work, in line with previous research — e.g. [14] [1]. However, in this article, the role of trust has not been explicitly explored as it poses a different perspective to the analysis — psychological rather than sociological. From our work, that takes a practice-based approach [16], it is visible that socialization helps to create trusting relationships among team members, in turn facilitating the negotiation and establishment of common practices. However, we have not proven the relationship, as we have not performed any psychological testing or controlled-experiment. We exclusively show that there is a relationship between socialization and metawork and the establishment of effective coordinative practices, both facilitated and supported by the usage of SoSo: a deeper analysis could be performed in future research relating the roles of social talks, metawork and trust.

Finally, the relationship between coordination mechanisms and communicative genres deserves further investigation, resulting in a theoretical framework for analysing and describing cooperative practices in GSD and deepen our understanding of the role of social protocols. In a previous work [7], we described the role of SoSo in the ecology of channels of an established team; in this work we have shown how the ecology of channels is established in a new team. Further develop the ecology of channels concept in a more comprehensive theoretical framework could provide theoretical underpinnings for future research that aims to analyse and describe not only the role of SoSo but how cooperative practices and are established in GSD teams.

9. CONCLUSIONS

In GSD, the lack of face-to-face communication poses challenges to the cooperation among remote teams: establishing and maintaining common practices appears to be a major issue. This paper shows that SoSo is a flexible channel that can enact different kinds of communicative genres and that supports coordination mechanisms, helping GSD team members to deal with the lack of face-to face communication. The development of effective coordination mechanisms depends on the successful initial metawork, which is related with the social cohesion of the team, through encouraging chats and social talks. Moreover, thanks to SoSo, it is possible to negotiate and establish common social protocols, thus to reach shared practices despite distance. Various kinds of SoSo have different roles in the diverse phases of the project: Wiki and Forum are a persistent repository for the knowledge shared; IM is a glue between other channels and is the media where things happen if they can not take place somewhere else, acting as a dispatcher for other channels and as the main channel for social talk; Forum and IM serve as channels for situated articulation, that support established coordination mechanisms.

10. REFERENCES

[1] B. Al-Ani and D. Redmiles. Trust in distributed teams: Support through continuous coordination. *Software, IEEE*, 26(6):35–40, 2009.

[2] M. Y. Brannen and J. E. Salk. Partnering across borders: Negotiating organizational culture in a german-japanese joint venture. *Human relations*, 53(4):451–487, 2000.

[3] E. Carmel and R. Agarwal. Tactical approaches for alleviating distance in global software development. *Software, IEEE*, 18(2):22–29, 2001.

[4] M. Cataldo, M. Bass, J. D. Herbsleb, and L. Bass. On coordination mechanisms in global software development. In *Global Software Engineering, 2007. ICGSE 2007*, pages 71–80. IEEE, 2007.

[5] E. Ó. Conchúir, P. J. Ågerfalk, H. H. Olsson, and B. Fitzgerald. Global software development: where are the benefits? *Communications of the ACM, 2009.*

[6] D. Damian, L. Izquierdo, J. Singer, and I. Kwan. Awareness in the wild: Why communication breakdowns occur. In *International Conference on Global Software Engineering, (ICGSE), 2007.*

[7] Y. Dittrich and R. Giuffrida. Exploring the role of instant messaging in a global software development project. In *International Conference on Global Software Engineering (ICGSE), 2011.*

[8] Y. Dittrich, D. W. Randall, and J. Singer. Software engineering as cooperative work. *Computer Supported Cooperative Work (CSCW)*, 18(5):393–399, 2009.

[9] K. Ehrlich and M. Cataldo. All-for-one and one-for-all?: a multi-level analysis of communication patterns and individual performance in geographically distributed software development. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 945–954. ACM, 2012.

[10] M. S. Elliott and W. Scacchi. Free software developers as an occupational community: resolving conflicts and fostering collaboration. In *Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*, pages 21–30. ACM, 2003.

[11] R. Giuffrida and Y. Dittrich. Empirical studies on the use of social software in global software development-a systematic mapping study. *Information and Software Technology*, 2013.

[12] J. D. Herbsleb. Global software engineering: The future of socio-technical coordination. In *2007 Future of Software Engineering*, pages 188–198. IEEE Computer Society, 2007.

[13] J. D. Herbsleb and A. Mockus. An empirical study of speed and communication in globally distributed software development. *Software Engineering, IEEE Transactions on*, 29(6):481–494, 2003.

[14] S. L. Jarvenpaa and D. E. Leidner. Communication and trust in global virtual teams. *Journal of Computer-Mediated Communication*, 3(4):0–0, 1998.

[15] A. M. Kaplan and M. Haenlein. Users of the world, unite! the challenges and opportunities of social media. *Business horizons*, 53(1):59–68, 2010.

[16] D. Nicolini, S. Gherardi, and D. Yanow. *Knowing in organizations: A practice-based approach.* ME Sharpe, 2003.

[17] W. J. Orlikowski and J. Yates. Genre repertoire: The structuring of communicative practices in organizations. *Administrative science quarterly, 1994.*

[18] I. Oshri, J. Kotlarsky, and L. P. Willcocks. Global software development: Exploring socialization and face-to-face meetings in distributed strategic projects. *The Journal of Strategic Information Systems, 2007.*

[19] K. Rönkkö, Y. Dittrich, and D. Randall. When plans do not work out: How plans are used in software development projects. *Computer Supported Cooperative Work (CSCW)*, 14(5):433–468, 2005.

[20] K. Schmidt and L. Bannon. Taking cscw seriously. *Computer Supported Cooperative Work (CSCW)*, 1(1-2):7–40, 1992.

[21] K. Schmidt and C. Simone. Coordination mechanisms: Towards a conceptual foundation of cscw systems design. *Computer Supported Cooperative Work (CSCW)*, 5(2-3):155–200, 1996.

[22] A. Sigfridsson. The purposeful adaptation of practice: an empirical study of distributed software development. *PhD Thesis*, 2010.

[23] L. A. Suchman. *Plans and situated actions: the problem of human-machine communication.* Cambridge university press, 1987.

[24] J. Yates and W. J. Orlikowski. Genres of organizational communication: A structurational approach to studying communication and media. *Academy of management review*, 17(2):299–326, 1992.

A conceptual framework to study the role of communication through social software for coordination in globally-distributed software teams

Rosalba Giuffrida

IT University of Copenhagen - Software and System Section
Copenhagen, Denmark rogi@itu.dk

Yvonne Dittrich

IT University of Copenhagen - Software and System Section
Copenhagen, Denmark ydi@itu.dk

Abstract

Background: In Global Software Development (GSD), the lack of face-to-face communication is a major challenge and effective computer-mediated practices are necessary. Communication through Social Software (SoSo) supports team coordination, helping to overcome geographical distance; however, in literature, there is no conceptual tool for researchers to describe the role of SoSo within the ecology of channels used by GSD teams.
Objective: This paper presents a novel conceptual framework to study the role of communication through SoSo for coordination in GSD. By highlighting how communicative and coordinative practices are constituted and maintained, the importance of common social protocols is highlighted.
Method: The framework is based on the concepts of communicative genres and coordination mechanisms and it is motivated and explicated through examples from two qualitative empirical cases.
Results: This paper presents a novel conceptual framework that integrates and extends the analytic tools of coordination mechanisms and communicative genres, showing how they mutually support each other. Thanks to the framework, it is possible to highlight how SoSo supports team members in establishing, developing and maintaining social protocols during the remote cooperation; SoSo also supports collaborative Software Engineering tools, which provide templates for coordination mechanisms. In particular, the importance of social talks supporting team building through SoSo is highlighted as not allowing for social talk may turn out to be expensive.
Conclusions: This paper presents a novel conceptual framework to study the role of communication through SoSo for coordination in GSD. The usefulness of the framework is supported by empirical findings on the role of SoSo. The theoretical framework can be beneficial for future research that aims to analyze and describe not only the role of SoSo, but also how communicative and coordinative practices are established and maintained in GSD teams.

Keywords: Global Software Development, Communicative Genres, Coordination Mechanisms, Social Software, Human Aspects

1. Introduction

Software Engineering (SE) is a cooperative work [1] and software developers must coordinate their individual activities with tasks performed by other team members [2] in their everyday practices. Coordination relies on communication - direct communication as well as communication mediated by code, documentation and artifacts. Communication is fundamental to coordinate the cooperative work and to establish and maintain effective coordination mechanisms [3]. In Global Software Development (GSD) settings, effective coordination is challenging [4] due to the lack of face-to-face communication [5]. Research in GSD aims to overcome this challenge improving processes and tools for supporting cooperation among remote teams - see e.g. [4], [6], [7]. However, there is often a gap between predefined SE processes and models and everyday practices of software teams.

Communication through Social Software (SoSo) has been promoted to support distributed teams, helping to overcome geographical distance: for example, Instant Messaging (IM) can replace planned or impromptu face-to-face meetings [8], [9], that are not feasible in GSD settings; Wiki can be used for knowledge sharing [10]; microblogs can generate virtual watercooler conversations and can be used as an informal communication channel [11]. Further examples are reported in a systematic mapping study performed by the authors of this paper that also shows that very little research focuses on the usage of SoSo in GSD [12]. Often SoSo is not acknowledged as being fundamental for GSD teams by researchers and practitioners, that investigate other tools and practices for overcoming the lack of face-to-face communication - e.g. reducing intensive collaboration [5], increasing formal documentation [13] and working on organizational factors such as processes, structure and goal alignment [14]. However, the success of agile processes in GSD projects, points to another direction, as agile methods depend on close collaborations and frequent informal face-to-face communication, rather than lengthy documentation. Some studies suggest that agile practices mitigate GSD challenges: for example, Holmstrom et al. [15] found specific agile practices to be useful for reducing communication, coordination, and control problems, while Layman [16] proposes that agile methodologies dependent on informal communication can be used on GSD projects, despite geographic, technical, temporal and linguistic hurdles. These premises motivate the necessity of further studies on tools and practices in GSD, as no standard recommendations are yet available in the field. The use of SoSo in GSD practices might be more wide spread than what is visible in actual research publications [12]. The real challenge lies in answering the question about why and how SoSo can provide useful communication channels for distributed collaboration. Some indications are provided in [12] and by the success of many open-source projects (OSS) which are coordinated through the wide use of SoSo, such as Wikis, Forum and Instant Messaging [17].

In literature, there is no tool for researchers to conceptualize and investigate the role of SoSo communication for GSD coordination, thus its central role in the actual practices of GSD teams is challenging to prove. However, SoSo is part of the ecology of channels [18] used by software engineers [19] in their everyday practices. Thus, it appears promising to develop suitable conceptual tools to support the investigation of the role of SoSo in relation with other communication channels and with collaborative tools traditionally used by software engineers. This paper proposes a novel conceptual framework for analyzing and describing the role of SoSo in GSD, motivated and explicated through examples taken from two empirical cases. The framework is based on the theoretical concepts of communicative genres [20] and coordination mechanisms [3]. The analysis of communicative and coordinative practices of the two cases shows that SoSo is especially useful to allow team members to establish, develop and maintain social protocols - a set of rules, conventions, policies shared by people involved in the cooperative activity [3]. In this context, the role of SoSo is to support communication and its key function is complementing collaborative SE tools, which provide templates for coordination mechanisms. In particular, the importance of team building chats through SoSo is highlighted as not allowing for social talk may turn out to be expensive. The theoretical framework can be beneficial for future research that aims to analyze and describe not only the role of SoSo, but also how communicative and coordinative practices are established and maintained in GSD teams.

2. Related Works

Global Software Development (GSD) means splitting the development of the same product or service among globally-distributed sites [17]. There are many potential benefits that can arise from GSD, that is promoted to lower development costs due to salary savings and to decrease development time due to time-zone effectiveness, to reduce time to market and to access the most talented developers [21]. Developing software as a team is a challenging task, but developing software as a global team is even more challenging due to distance [13]. In particular, main challenges in communication and coordination arise due to the lack of face-to-face communication. Wang et al. [22] theorize a conceptual framework that aims to explain how some contextual dimensions of GSD influence practitioners' attitudes toward, and usage of, the tools; one of the five dimensions identified is inter-team communication and coordination [22]. However, in their work, Wang et al. do not indicate which conceptual tools can be used to study and describe this dimension. Scacchi [23] refers to the concept of "Software informalisms", which are the information resources and artifacts that participants use to describe, proscribe, or prescribe what is happening in a OSS project. Software informalisms are the media and the subject of software requirements/design. Reading, reviewing, and reinterpreting informalisms is a prerequisite to writing OSS. The artifact-based approach proposed by Scacchi appears advisable to study likewise GSD communicative and coordiantive practices, as in GSD, digital artifacts are used for most of the distributed collaboration; therefore, an artifact-based perspective is kept in the following. However, the notion of software informalisms regards artifacts both as a medium and as the subject, while the two aspects deserve specific considerations, especially if we aim to understand the role of SoSo - as a communication channel - for coordination in GSD.

McChesney and Gallagher [24] proposed a framework to describe coordination activities in SE. In their article, McChesney and Gallagher relate coordination theory [25] and communication genres [20] and use the framework to explain and interpret the complex web of personal interactions observed in two real-word software projects. They argue that "there is a gap between existing process-oriented method for describing software processes and the situated activities in which software engineers engage when developing systems"; thus, a better understanding of actual coordination practices is necessary to bridge this gap. Following McChesney and Gallagher approach, this paper likewise aims to analyze communicative and coordinative in situated action [26] and how methods and processes are adapted by team members. However, though McChesney and Gallagher attempt to analyze both coordinative and communicative practices in SE, they stick to coordinative practices, leaving the relationship between coordination and communication unexplored. They quickly abandon the concept of "coordination mechanisms" as proposed by Schmidt and Simone [3] to replace it with the "communicative genres" notion by Yates and Orlokowski [20]. In our field material, "coordination mechanisms" and "communicative genres" appear as complementary concepts, both necessary to understand distributed software practices, and as compatible concepts, both based on the notion of social protocols. Thus, an elaboration of the framework appears necessary.

In SE as well as in GSD literature, the concept of "coordination mechanism" is widely used to indicate a mix of a broad set of practices, methods, processes and tools, as the "mechanism" concept is frequently used as a general term not related to actual SE practices. In particular, coordination is defined in the Coordination Theory proposed by Malone and Crowston [25] as "managing dependencies between activities"; in this context, coordination is achieved by one or more "coordination mechanisms": each one addresses one or more dependencies in a situation. Studies in SE as well as in GSD adopt Coordination Theory to investigate software development activities, see e.g. [4, 27, 28]. In particular, in the context of an agile software team [29], tools such as the wiki and the product backlog, activities as the daily standup meeting, and roles such as the project manager are all considered coordiantion mechanisms. Similarly, in the context of a GSD project [14], a set of coordination mechanisms of various nature are analyzed, such as centralized team structure, documentation, periodic commit, communication tools and periodic meetings. Schmidt and Simone agree with Malone and Crowston [25] that cooperative work arrangements have to cope with inter dependencies of different complexity. However, empirical evidences provided by Schmidt and Simone in the context of cooperative work research, show the widespread use of coordinative practices that rely on coordiantive artifacts. Thus, as part of the actual CSCW practices and based on the use of artifacts for coordination purposes, "coordination mechanisms" are defined by Schmidt and Simone with a rigorous definition that encompasses concepts such as social protocols and articulation work, that appear promising to understand coordination in GSD, using a conceptual tool that allows to explain how different practices occur in different projects and to compare them. For these reasons, in the framework proposed in this paper - that is an elaboration of the framework proposed by McChesney and Gallagher [24] - the concept of "communicative genres" notion by Yates and Orlokowski [20] is used in combination with the concept of "coordination mechanisms" proposed by Schmidt and Simone [3]. A detailed description of both concepts is reported in Section 5 and the two concepts are explicated and extended through several examples taken from two empirical cases of GSD teams.

3. Description of the Cases

Examples used for explicating the theoretical framework proposed in this paper are taken from two empirical cases: DHI, an industrial case, and the case of three GSD student teams. A brief description of the two cases is reported in the following.

3.1. An Industrial Case: DHI

DHI is an independent, international, consulting and research organization. The company develops and uses high-end hydraulic simulation software. We have observed part of the World Bank Project (WB-Project), which has a considerable amount of software development. This is a global distributed project: five members are settled in Copenhagen, Denmark; seven members in Delhi, India and one Project Area Manager in Portland, USA. The Danish team is composed of one project manager and four Project Area Managers (PAM), the Indian team consists of five developers, one team leader, and one tester. We observed the team while working on the development of a Decision Support System (DSS) for water management in the Nile Basin. The observations took place during the final part of the development process of the first release of the system mainly during the testing phase of the same release. The project was successful, and the team has subsequently developed a second release of the software. The observations took place both in Copenhagen and in Delhi in Fall 2010; the first author observed the team for four months in Denmark and for two weeks in India. The tool used by the team is Spira[1], a Test Management tool, used by the team as an issue tracking system. All development activities are tracked in the system: the description of features to develop, reporting of incidents, assignment of tasks, and description of test cases. Spira automatically assigns an "incident number" to all defects, test cases and requirements. Skype[2] is the team's main tool for communication and team members are supposed to have it switched on when they are at work. Team members use different Skype channels for synchronous or nearly synchronous collaboration: written IM, audio, and screen sharing. Usually, one-to-one Instant Messaging (IM) chats take place through Skype.

3.2. The three Student Projects Case

The three projects under study are part of a GSD student cluster in collaboration between IT University of Copenhagen (ITU), Peking University (PKU) and Universidade Federal de Pernambuco (UFPE). Each team is formed by two remote subteams. One team (Team A) has participants located in Denmark and in Brazil, while the other 2 teams (Team B and Team C) have participants located in Denmark and in China. The collaboration took place from February 2011 to May 2011. An academic supervisor provided the description of the product to be developed and he evaluated the work performed by the students based on the code developed, on a final report produced by each sub-team and on an oral exam. The system design, the requirement specifications, the development of the product and the organization of the collaboration were students' responsibilities. Therefore, the teams were "self-organising" their work, sharing roles, responsibilities and decision taking.

Tools used by the three teams are: Skype, for video conference and IM chats, emails and Assembla[3], an issue tracker system with additional functionalities such as file sharing, Wiki, Forum and an integration to Subversion (SVN) [4], the version control system used. Most of the collaboration took place in Assembla and during a weekly meeting of about one hour through Skype. Assembla was used for keeping track of the status of the project, for managing issues, for defining deadlines and as a shared repository for file exchange. Email were rarely used, the forum of Assembla substituted the email for communication among team members: it worked as a common repository of the

[1] `http://www.inflectra.com/HomePage.aspx`

[2] `http://www.skype.com/`

[3] `http://www.assembla.com/`

[4] `http://subversion.apache.org/`

messages exchanged by team members. Moreover, it was possible to enable the notification via mail functionality that updated team members via mail about the new events of Assembla. IM chats were rarely used due to time zone differences and lacking of overlapping working hours. A Skype group chat with all team members has been carried on during the whole project by Team C and it has been mainly used in the initial phases of the project. All teams succeeded in having a working prototype. The collaboration is considered successful or not based on the self-reported impressions of team members in the final report and during the observations by the researcher.

4. Method

This paper explicates the novel conceptual framework through examples taken from two ethnographically-inspired studies: an industrial case and a three student projects case, both described in the previous section.

4.1. Data Gathering

The two real world cases have been observed and analyzed through different data collection techniques: non-participant observations, semi-structured interviews and document analysis. In order to carefully take track of the investigation, a research diary has been kept during each project observed, meetings and interviews have been taped and transcribed, the content of computer-mediated communication through SoSo has been analyzed both qualitatively and quantitatively. By using non-participant observations, where the first author participated as a researcher in the daily routines of the teams, it was possible to observe how collaboration took place between team members in the different projects. Semi-structured interviews gave the possibility to clarify uncertainties and to ask about specific issues in a deeper way. Interaction analysis [30] was performed for analyzing SoSo logs. Workshops were organized in the industrial case to summarize the outcomes and to support researchers and practitioners in reflecting together on the findings obtained. By using multiple ways of collecting data and combining different kinds of methods, it has been possible to triangulate the findings [31].

4.2. Data Analysis

Previous research of the authors reported the analysis of the role of Instant Messaging in the DHI case [18] and the analysis of communicative and coordinative practices of one of the student teams [32]. The theoretical framework presented in this paper has been developed based on the field material offered by both empirical cases. However, while in the first paper [18], an interaction analysis of the chats has been performed based on their communicative purpose and no conceptual tools have been adopted, in the second paper [32] the concepts of communicative genres and coordination mechanisms have been used, showing a promising relationship between the two concepts, that deserves further investigation. Therefore, in this paper, the relationship between communicative genres and coordination mechanisms is further detailed, using the theoretical concepts to re-analyze part of the field material of the DHI case and to extend the analysis of the student team material with the two teams not studied in [32]. The theoretical framework is used for describing and analyzing practices, meanwhile it is motivated and explicated by the empirical material.

5. Developing the Theoretical Concepts for the Framework

This section describes in detail the concepts of *coordination mechanisms* and *communicative genres* that are the basis for the theoretical framework presented in Section 7 and that are illustrated with examples taken from the empirical cases presented in Section 3. Building blocks of the two concepts are reinterpreted from the original definitions to provide a comprehensive understanding of the relationship between communicative and coordinative practices. Thus, the definitions of *communicative purpose*, *articulation work*, *social protocols* and *genre repertoire* are extended and discussed in details in the second part of this section, providing additional examples.

5.1. Coordination Mechanisms

5.1.1. Definition

The concept of *coordination mechanisms* is defined by Schmidt and Simone [3] as follows:

> "A *coordination mechanism* consists of a *coordinative protocol* imprinted upon a distinct *artifact* which [...] *stipulates* and *mediates* the articulation of cooperative work so as to reduce the complexity of *articulation work* [...]" [3] (emphasis in original)

The coordinative protocol consists of a set of rules - e.g. taken-for-granted ways of proceedings, established conventions, official policies, standard operating procedures. The coordinative artifact is a "stable data structure expressed in a standardized graphical format" [3]. Schmidt and Simone report, for example, about a bug report form[5], a two page form (the artifact) with several fields filled by different actors, which follows a set of agreed procedures and conventions (the protocol) and which stipulates the responsibilities to the different roles, the possible classifications of bugs, reports of bugs corrected, etc. The artifact is "the distinct and persistent symbolic construct in which the protocol is imprinted and objectified" [3]. In cooperative work settings characterized by complex task interdependence, "coordination mechanisms reduce the complexity of articulation work and alleviate the need for ad-hoc deliberation and negotiation" [3]. Since SE is cooperative work [1], the concept of coordination mechanism can be beneficial when studying GSD coordinative practices.

5.1.2. Extending the definition

The concept of coordination mechanism implicitly comprises the notion of purpose that is explicitly introduced in our theoretical framework. The purpose of the coordination mechanism

[5] The article was published before issue tracker were wide spread in industry.

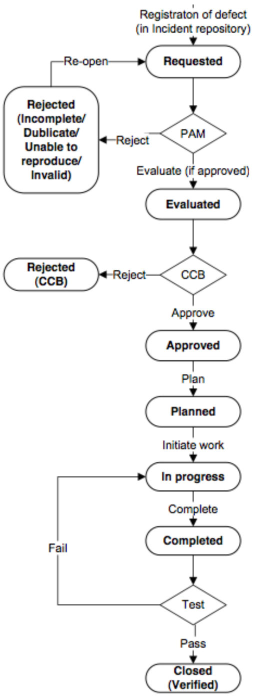


Figure 1: Incident Workflow

is indicated in the name of the coordination mechanism itself. For example, an issue tracker mechanism allows the tracking of issues, while a file sharing mechanism has the purpose to share files. Further examples of coordination mechanisms are described in the following.

5.1.3. Examples

Many coordination mechanisms are used by software teams when developing software. Anytime an artifact is used to coordinate the cooperative work and team members agree on a social protocol about how to use it, an effective coordination mechanism is established. Coordination mechanisms in GSD are generally constituted by digital artifacts, thus they are easy to share across distance. The establishment, development and maintaining of social protocols is generally more challenging in GSD due to the lack of face-to-face communication. In the following, two examples are presented, the first describes the incident coordination mechanism used in the DHI case, while the second explains the establishment of the agenda coordination mechanism in one of the student projects.

Incident Coordination Mechanism. In the DHI team, one main coordination mechanism has been identified [18]: the tracking of development activities through Spira, a Test Management tool, used as an issue tracker system. All development activities are tracked through Spira: the description of features to develop, the reporting of incidents, the assignment of tasks, and the description of test cases. Spira automatically assigns an "incident number" to all defects, test cases and requirements, that are the actual *artifacts* of the coordination mechanism. In the "Project handbook" document, part of the documentation available in the team, there is a diagram - reported in Figure 1 - that formally describes the workflow of an incident, thus it explicates the *social protocols* that team members need to share in order to allow the coordination mechanism to function. The process is clearly defined: a defect that has been registered - in Spira - has to be approved by the PAM, who can reject it or evaluate it. During the morning meeting, Danish PAM - Change Control Board (CCB) - usually approve and plan the incident, assigning it to a developer through Spira. When the developer starts working on the defect, he changes its status as "in progress", and when he finishes working on that, he marks it as "completed". He then assigns the incident to the PAM, who can test and verify it. The process defined in the diagram is supported by Spira, and the current status of every defect is visible in the tool at any time.

Agenda Coordination Mechanism. In Team C [32] of the student team case, the Danish sub-team started to share an agenda for the meetings from the very beginning of the collaboration; the agenda was shared during the meeting through the Group Chat. After some weeks, a problem was reported in a Forum message by the Chinese sub-team in reaction to a not successful meeting: the Chinese sub-team explicitly requested to share the agenda prior to a meeting:

> We, PKU, advise that you, ITU, tell us agenda in advance. In that case, we, PKU, have time to prepare for it. And our meeting will be better. O(∩_∩)O *[Wusheng, China]*

In the same week, the Danish sub-team produced a document about the collaboration, in which rules for the sharing of the agenda were proposed:

> A written meeting agenda should be uploaded at Wednesdays in order to accomplish effective virtual meetings. The author of these should be shifting between China and Denmark. So e.g. 1: one team post a agenda proposal on Wednesday 2: the other team confirm, write a respond to it or add things to the agenda latest 2 hours before the meeting on friday. 3: the team who originally created the draft will add a final agenda latest 1 hour before the meeting starts.

The suggested rules - proposals for social protocols - were discussed in a Forum thread and the coordination mechanism got established: after the agreement was reached, the agenda (*artifact*) was shared by the one sub-team through a Forum message, it was integrated by the other sub-team and finally modified by the sub-team that proposed it (*social protocol*). This practice was carried on alternatively by the two sub-teams similarly during the remaining of the collaboration.

In summary. This section reported two examples of coordination mechanisms: (1) the *incident mechanism*, an established coordination mechanism formally defined in the documentation of an established team, and (2) the *agenda mechanism*, a novel coordination mechanism that was established in a forming GSD team. In the former example, a collaborative SE tool - the issue tracker system - supports the coordination mechanism, while in the latter, the coordination mechanism is supported by communication through SoSo tools, namely IM group chat and Forum. In order to thoroughly describe how the communication occurring in GSD teams supports coordination mechanisms, the analytic tool of communicative genre is introduced and described in the following section.

5.2. *Communicative Genres*

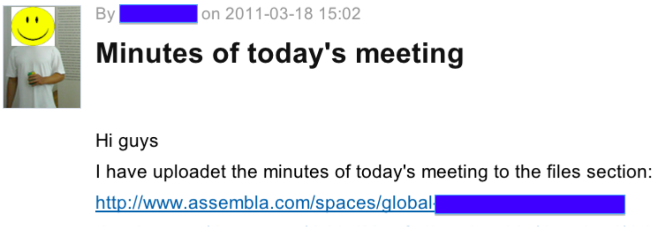
5.2.1. *Definition*

Genre Theory states that genres create order to simplify the mass of available information [33]. Yates and Orlikowski widely investigated the notion of *communicative genres* as a way for structuring practices in organizations [20]. A genre of organizational communication is a "distinctive type of communicative action characterized by a socially recognized communicative purpose and a common form" [34]. The communicative *purpose* of a genre is not rooted in a single individual's motive for communicating, but in a purpose that is constructed, recognized and reinforced within a community [35]. The *form* of a genre refers to the readily observable features of the communication, including structural features, communication medium and language [20]. *Structural features* can be text formatting, such as lists and headings, as well as devices for structuring interactions at meeting, such as agenda and chairperson; the *communication medium* can be pen and paper, face to face, a telephone or a mail; the *language* can be the level of formality or the specialized vocabulary used [20].

People produce, reproduce and change genres through a continuous process of negotiation and re-adaptation [34]. Genres may be considered at different levels of abstraction and they can be combined in *genre repertoire*, a set of genres routinely enacted by a particular community. A community's genre repertoire indicates "its established communicative practices and it can serve as a analytic tool for investigating the establishment of a community's communicative practices" [20]. Im et al. [36], for example, used genre and genre repertoire for analysing email communication of a geographically dispersed software team. In their work, they analyse and describe in detail the genres identified in the electronic communication among team members. It appears promising to adapt a similar approach and to perform communicative genre analysis also on other communicative channels used by distributed teams, such as SoSo.

5.2.2. *Example*

Many different communicative genres can be identified in our empirical cases. Genre analysis can be performed on mail, SoSo, but also on meetings and in general on any "type of communicative action characterized by a socially recognized communicative purpose and a common form" [34]. Our interest is in computer-mediated traces of communication, in particular in classifying the usage of SoSo, in order to understand its role in GSD practices. An example of communicative genre recurrent in two different projects of the student teams case is reported in the following.



By [redacted] on 2011-03-18 15:02

Minutes of today's meeting

Hi guys

I have uploadet the minutes of today's meeting to the files section: http://www.assembla.com/spaces/global[redacted]

Figure 2: Minutes Notifier Genre

Minutes Notifier. In the student teams case, the Minutes Notifier genre is recurrent in Team A and in Team C - an example of Minutes Notifier genre is reported in Figure 2. The communicative *purpose* of the genre is to inform team members that minutes of the weekly meeting have been uploaded in the file section of Assembla. The *form* of the genre is similar in all Minutes Notifier messages identified:

- the *communication medium* is the Forum available in Assembla;
- the *structural features* make the genre easy recognizable: in the title the purpose of the message is reported - "minutes of today's meeting"-, followed by a brief message and a link to the file uploaded;
- the *language* is usually based on specific terms - such as "minutes" and "file section" - and goes straight to the point, with a brief informal opening - "hi guys".

The Minutes Notifier genre has been developed within two of the student teams (Team A and Team C) with the purpose of making all team members aware that a specific file - i.e. minutes of the meeting - has been uploaded. An automatic functionality of Assembla enables a similar communication purpose - e.g. team members can receive an automatic mail whenever a file is uploaded. However, this practice has been established and, despite the coordination mechanism of file sharing is well supported by the file sharing system available in Assembla, an independent communicative genre is used to enact and reinforce the coordination mechanism. This appears to be a common relationship between communicative genres and coordination mechanisms, as it will be described in detail in Section 6.

Please not that the purpose of the Minutes Notifier communicative genre is to inform team members that minutes of the weekly meeting have been uploaded, thus to provide awareness to other team members. As we will show in Section 5.4, the awareness purpose underpins the situated articulation genre, that is a genre category that includes also the Minutes Notifier genre. Thus, a genre can be part of a genre category, sharing the purpose but differing in the form. The notion of purpose is further detailed in the next subsection.

5.3. *Communicative Purposes*

5.3.1. *Extending the definition*

In their analysis of communicative genres in mail, Orlikowski and Yates [20] identify purposes such as: response, question, proposal, for your information (FYI) and meta-comment. These purposes are not topic-related and are not sufficient for the detailed analysis of concrete interaction through SoSo. In our framework, the concept of *purpose* of a communicative genre is considered with a contextual meaning, rather than the one presented by Orlikowski and Yates in their studies. Since, as mentioned above, communication genres can be identified at different levels, we consider content related purposes - such as awareness and team building - in order to understand the roles of communicative genres within collaborative software development practices. The response/question dimension used by Orlikowski and Yates [20] is maintained as a generic categorization that, when appropriate, specifies sub-genres with more specific purpose within the content related purposes identified.

5.3.2. Examples

In the student teams project reported in [32], several communicative genres are identified: work discussions, knowledge sharing, encouraging chats, social chats, metawork and situated articulation. A detailed description of each communicative genre is outside the scope of this paper; however, few examples are reported in the following to clarify the perspective adopted in our framework and how it differs from Orlikowski and Yates [20] approach. In particular, in the following, examples of *Work discussions* genre and of *Team building* genre are reported, both from the student projects case and from the DHI case. *Metawork* and *situated articulation* genres deserve a detailed discussion, as they are related with the articulation work concept introduced in Section 5.1. This is detailed in the subsequent subsection.

Work Discussions. Work discussions have the purpose of collaborating with remote team members. Work discussions can be related with decision making, requirement specification, project planning, technical issues, etc. In our empirical cases, work discussions can be found in IM chats, in Forum and in mails, with slightly different structural features and level of formality of the language, due to the medium used, but with the same work purpose - thus they fall in the same genre category. Work discussions usually start with a question, followed by one or more answers, that sometimes lead to a broader discussion. The question/answer purpose suggested by Orlikowski and Yates [20] is present as subcategory to add details to the purpose of performing the actual work, discussing about solutions to implement or issues encountered. An example of work question in the Forum of one of the student teams is reported below:

> CinemaService URL??
> Hey Rafael! What's the URL I should use to initiate the cinemaService?? *[Tommy, Team A]*

In the example, the work question is reported in the title of the forum message and in the post itself. The question is about a concrete work issue. One team member, Rafael, is addressed, despite it is a Forum message broadcast to all team members. This way, if someone else in the team knows the answer, he can reply to the question; moreover, being in a Forum thread, the information is stored in a more persistent repository than e.g. an IM chat.

In the DHI team, work discussions have also been highlighted in [18] while analyzing IM chat logs and have been classified as collaboration chats. However, as reported in Section 4.2, the re-analysis performed in this paper through the communicative genre concept allows to show that the chats categorized under the purpose of collaboration in [18] fall in the genre category of work discussions. An example of IM chat of the DHI case is reported in the following.

> *Jakob:* Does your solution for 699 hide any field that is of type GUID? The main reason this came up is because when you do an Intersect or Union, the GUID from the two input feature classes are added to the output table. This is what Joe wanted hidden, not just the standard "ID" field.
> *Naveen:* oh okay, I didnt do that :(
> *Jakob:* Yes, I think the criteria should be if the field type is "Guid", not what the field name is.
> *Naveen:* but cant some other guid fields could be important.
> *Jakob:* Possibly, but for now we want them all hidden.
> *Naveen:* okay will do that.

In this example, it is possible to see how work discussions chats complement formal design specifications. Team members refer to the incident number (699) and discuss about the solution of an incident, clarifying what the software should do under specific conditions. The work discussion starts with a question and the technical issue is solved with very fast interactions between co-workers, probably faster than detailing it in formal design documents. Interestingly, managers agree on the fact that sometimes it is easier to have a direct discussion about a specification through IM chat rather than detailing it formally through documentation and that fast chats like the one reported in the example overcome the lack of face-to-face communication. However, sometimes the decisions occurred in the chat can be reported in the issue tracker system, in order to keep track of the specifications. When the topic is too complex for a written chat, the conversation started in IM can switch to a call. Moreover, if issues are considered to have a wider implication, team members may start a discussion on mails [18].

Team Building. In the analysis of the student project communicative genres [32], we identified two two different genres with the purpose of team building: socialization genre and encouraging chats. These genres can not be related to any question/answer purpose proposed by Orlikowski and Yates [20], however, they appear fundamental in communication among team members, both as an explicit purpose or as a sub-text [18] in other genres - e.g. work discussions.

Socializing chats, referred also as cheap talk [37], consist in messages with the sole purpose of socializing with the remote team members, talking, for example, about each others' country; an example of IM chat occurring in the Team C of the student project is reported below:

> *Zhang:* Denmark is a beautiful country. I will go to Denmark to have a tour if there any chance, To got to know this country
> *Stella:* well please come... It is beautiful!! I have several chinese colleagues and friends at work (sushi restaurant)

This chat occurred in the beginning of the collaboration, when team members used IM chat to get to know each others and try to establish a good working environment.

In the DHI team, pure socialization chats are rare as team members know each other. An example of socialization chat in the DHI case is reported in the following, that is an introductory socializing talk occurring at the beginning of the day as a "good morning" that helps to establish a channel for the collaboration that follows.

> *Prashant:* Hi Jonas
> *Jonas:* hi Prashant
> *Prashant:* hows monday treating you?
> *Jonas:* better than usual. Working from home
> *Prashant:* oh we r on the same boat
> *Jonas:* good to hear :)

The socialization chat provides awareness about the situation of the remote colleague Jonas, with whom Prashant is used to work daily; the chat allows to establish a channel for further communication during the day.

Encouraging messages are positive feedback to a message posted by another team member, which can belong to any other genre. Two examples from the student projects are provided below:

> You did a great job! 佩服你们。 And we will finish our part of self-introduction soon. *[Yang, Team C]*

> This is really cool:] *[Arnold, Team B]*

The two examples clearly show appreciation for the work performed and encourage remote team members to continue the successful collaboration. In the first case, some Chinese characters are reported as a translation of the English sentence "You did a great job", adding a funny ad challenging connotation to the message. Encouraging chats in the student teams happen both in Im and in Forum.

In the DHI team, encouraging IM chats are very frequent. In the following example, after a technical discussion, the manager gives encouraging feedback to the developer and shows that he appreciates the work that has been done.

> *Julius:* uh - you are very methodic. That is great
> *Shahid:* Thanks
> *Shahid:* but if i have missed any..then please let me know
> *Julius:* I will - but please home now :)

The example shows Julius' appreciation for the work performed by Shahid and the commitment of Shahid. The tester is working until late in the evening to finish his job and he wants to be sure he has done all tests necessary - "if i have missed any..then please let me know". Julius assure Shahid and pushes him to "go home" acknowledging the commitment of the tester. These encouraging chats allow team members to maintain a good working relationship despite geographical distance. The importance of encouraging chats is highlighted by one of the PAM of the DHI case:

> Our team is not so big, so I think it's important to take care about what is happening and give the right attention to each team member. *[Jakob, Denmark]*

Summarizing, socialization chats are visible, mainly through IM, in the initial phase of the student project and as an initiation of the working day in the industrial case. Encouraging chats are common both in the student case and in the industrial case, through IM or Forum messages, and are usually reactions to successful achievements of the remote team members. Depending on the maturity of the team, different kinds of team building conversations can happen and they seem to establish, support and improve the collaboration across sites. The supportive role of team building chats for the development of other communicative genres and for the success of the collaboration is further discussed in Section 6.4.

5.4. Articulation Work

The articulation work concept has been introduced in Section 5.1, in relation with the definition of coordination mechanisms. However, articulation work can be understood as a communicative genre categories. Thus it deserves a particular attention and an extension of the definition is provided in the following, supported by some examples.

5.4.1. Extending the definition

Schmidt and Simone refer to the concept of *articulation work*, that is defined as:

> "[...] a recursive phenomenon in that the management of an established arrangement of articulating a cooperative effort may itself be conducted as a cooperative effort which, may also need to be articulated." [3]

In cooperative work settings characterized by complex task interdependence, "coordination mechanisms reduce the complexity of articulation work and alleviate the need for ad-hoc deliberation and negotiation" [3]. In our framework, two levels of articulation work are distinguished, as suggested by Gerson [38] and Strauss [39]: metawork and situated articulation. *Metawork* is used to describe the development of a social protocol. *Situated articulation* denotes the discussion of the state of the current task in order to coordinate this task; situated articulation involves "adapting a social protocol to a situated use" [40].

Metawork and situated articulation can be understood as categories of communicative genres that are used for changing and articulating not only the coordination mechanisms, but also the communicative genres of a project team.

5.4.2. Examples

Despite coordination mechanisms should alleviate articulation work, it will become visible that articulation work - in the form of metawork - is necessary to establish coordination mechanisms and that, once the coordination mechanism is established, further articulation work - in the form of situated articulation - appears fundamental to support the coordination mechanism. This section provides some examples to show the difference between the two kinds of articulation work, while Section 6 will detail the supportive role of metawork and situated articulation for establishing and enacting coordination mechanisms.

Metawork. Metawork communication consists in metacomments about how to structure the work within the team. They can be proposals, questions, answers or discussions about how to carry on the work. Metawork allows to establish coordination mechanisms. An example of metawork proposal occurred in the Forum of one of the student teams is reported below:

> Hello, everyone of ITU, I made a table by Word. In order to find the best time for our meeting, please download the .doc file and fill the table, and then upload it in a reply message. By it, we can have a clearly view which is the proper time. Thanks! *[Cheung, Team C]*

In the example, Cheung suggests a procedure to follow (*social protocol*) for using the word document (*artifact*) to find the best time for the weekly meeting. Thus, a temporary coordination mechanism is established by Cheung that in the Forum metawork message suggests the social protocol to follow.

Metawork is supported both by IM chats and Forum messages and helps to establish persistent coordination mechanisms, as shown in detail in the temporal analysis performed for Team C and reported in [32]. Thanks to metawork, effective negotiation of social protocols and decisions about coordination mechanisms have been taken among team members and metawork discussions usually end when coordination mechanisms get established.

In the DHI case, metawork does not appear in the IM conversations: team members just did things, without discussing about how the work had to be performed. This is probably related to the fact that the team is established and coordination mechanisms are working effectively. It could be possible, however, that metawork takes place in other channels than IM, in order to inform all team members of practices to follow, e.g. in mails. However, we did not analyze in detail the mail communication of the DHI team, thus only hypothesis can be suggested.

Situated Articulation. Another kind of articulation work, situated articulation, is visible in communicative practices of both industrial and student case. Situated Articulation communicative genres are questions, answers or FYI messages about the work that has been done so far, with the purpose to provide awareness to all team members about the status of the project. In particular, team members can ask for an update about the status of the project, they can reply or independently provide an update of what they did recently. The Situated Articulation genre is particularly interesting because it is used for supporting and enacting established coordination mechanisms.

In the three student projects, a large amount of the *File Notifier* genre is reported as an example of situated articulation genre. The *File Notifier* genre is a broader genre that includes the *Minutes Notifier* genre described in Section 5.2.2. File Notifier messages consist in notifications about the sharing of files, of minutes, of the agenda and of the standup meeting that support the associated coordination mechanisms. It is interesting to note that in Team C, once coordination mechanisms get established, patterns of collaboration are clearly visible in the Forum, e.g. as sequences of messages: standup, agenda, minutes [32].

In the DHI case, many situated articulation chats are visible. In [18], the authors referred to these chats as coordination chat; however, the coordination dimension is present in the coordination mechanism itself, while the situated articulation chat has the purpose to provide awareness, supporting the coordination mechanism. An example of situated articulation chat, in which the social protocol is "adapted to a situated use", is reported below.

> *Prashant:* Hi Jonas..
> *Prashant:* I have implemented [...]
> *Prashant:* the implementation is working fine
> *Prashant:* do u want me to check-in the code and test on ur system??
> *Jonas:* maybe you could show it to me ?
> *Jonas:* I'll call when Morten is ready
> *Prashant:* ok.call me once you guys are ready

In this example, Prashant reports that he has completed a task and he makes Jonas aware that the "implementation is working fine". The fact that the task is completed is reported in Spira, the issue tracker system, and, following the incident workflow - see Section 5.1 -, Jonas should check the implementation without the explicit request of Prashant. The coordination mechanism is established in the team and theoretically it does not require further communication. However, since the incident is of particular relevance, Prashat wants to remark the finalization of the task and he wants to be sure that the implementation follows Jonas' expectations. Therefore, a situated articulation message is sent to Jonas to adapt the social protocol defined for the coordination mechanism to the specific case, highlighting the importance of the incident solved. A broader discussion on the establishment, adaptation and maintaining of social protocols is carried on in the following subsection.

5.5. Social Protocols

5.5.1. Extending the definition

The definition of coordination mechanisms - see Section 5.1 - explicitly mentions the role of the social *protocol*: "a set of rules, conventions, policies shared by people involved in the cooperative activity" [3]. In the communicative genres definition, protocols are not defined; however, it is reported that *social norms* are visible in recurrent communicative situation [34], thus, social norms are underpinning also the concept of communicative genres. More in general, the concept of *social protocol* (or social norm or social rule) is related with the concept of *practice* and in particular with the social nature of it, as stated by Wittgenstein [41]:

> [...] *obeying a rule* is a practice. And to think one is obeying a rule is not to obey a rule. Hence it is not possible to obey a rule *privately*: otherwise thinking one was obeying a rule would be the same thing as obeying it. (P.I. 202 [41])

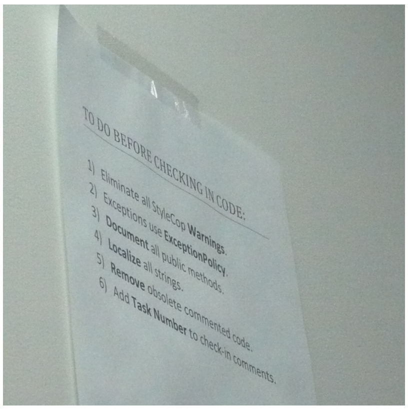
Thus, the social nature of the norm comprises the necessity to be shared by people involved. Wittgenstein also suggests that [41]:

> [...] we lay down rules, a technique, for a game, and that then when we follow the rules, things do not turn out as we had assumed (P.I. 125 [41])

As to say, social protocols not only need to be initially *decided*, but also *adapted* and *adopted* by people during time, thus socially *shared*, *modified* and *appropriated*. This phenomenon is often referred as local and temporary alignment of social practices - see e.g. [42], [43]. From our field material, social protocols appear to come about and to be maintained in different ways. Social protocols can be both explicitly and implicitly defined by team members. A social protocol is *explicitly* defined through metawork, when team members discuss whether and how to use a specific artifact. For example, a team can decide to share an *agenda* (the artifact of the coordination mechanism) prior to a meeting and that each sub-team is alternatively responsible for that. A social protocol can be *implicitly* adopted e.g. based on previous *work experience* in the project or from *professional knowledge*. The social protocol on how to use the agenda during the meeting might be *implicitly* driven by *previous experience* of team members and does not require further discussions on how to use the artifact. A social protocol can also be *defined* but *not adopted*, when team members agree on it, but later do not use it. Social protocols are formed, negotiated, established and maintained by team members in GSD teams also thanks to the usage of SoSo. Some examples from the DHI case are reported in the following.

5.5.2. Examples

In the DHI case, both Word and PDF documents are used to formalize the planning and the execution of the WB project. Project Organization and Project Life Cycle are formally defined in a document with guidelines for the Software Development Project. The document incorporates also a short, step-by-step guide on how Spira should be used by the team to manage software development projects. The social protocol on how to use Spira is formally described in the documentation, thus it is defined, shared and established among team members. From the observations performed during the project, the researcher can confirm that the social protocol is also adopted by team members. Despite the social protocol is well-established, some IM chats are present in the field material that show how social protocols need to be maintained among team members. An example of situated articulation chat is reported in the following.



TO DO BEFORE CHECKING IN CODE:

1) Eliminate all StyleCop **Warnings**.
2) Exceptions use **ExceptionPolicy**.
3) **Document** all public methods.
4) **Localize** all strings.
5) **Remove** obsolete commented code.
6) Add **Task Number** to check-in comments.

Figure 3: Todo before checking the code

> *Naveen:* any incident you have planned for me...
> *Naveen:* to fix
> *Jonas:* look in SPIRA for planned - take from the top with respect to priority

Even though the social protocol is known by Naveen, he explicitly asks Jonas if there is something in particular he should work on, to double check whether to rely on Spira - thus on the coordination mechanism - or not. Jonas answers confirming to follow the social protocol, that through this chat is reinforced and maintained.

In the documentation, the role played by the "incident number" assigned by the tool to any issue inserted in the system is formally described. However, the number is also used in the check-in comments of the source code, as defined - in a less formal way - in a to-do list available in a piece of paper hanging on the wall of the the Indian site - see Figure 3. In this case, the social protocol on how to use the issue number in the check-in comments is defined - less formally -, shared, established and adopted by team members.

Another social protocol underpins the use of the incident numbers. Team members continuously refer in the IM chats to the incident number, as shown in the following example.

> [10 : 07] *Arun:* 974 is yours now for verification (gap filler scrollbars)
> [10 : 09] *Nelson:* thanks
> [11 : 58] *Arun:* 1008

[12 : 07] *Nelson:* all yours
[12 : 57] *Arun:* your now.
[13 : 18] *Nelson:* 847 is yours

In this case, a situated articulation chat is carried on between the two team members that exchange incident numbers in the chat, along with assigning the task in Spira, and the incident numbers are bouncing back and forth, allowing fast coordination among team members. In the example, time stamps are reported and show that collaboration took place during the whole morning; team members relied on each other to know what to do and a shared social protocol allow them to understand what to do when they receive an IM chat with a number. The situated articulation chat augments the coordination mechanism enacted in Spira, while the social protocol - undefined, implicitly adopted based on previous work experience and shared - allows team members to understand the meaning of the number mentioned in the chat.

5.6. The Repertoire

In this subsection, the concept of repertoire, defined for communicative genres, is described and adapted to the set of coordination mechanisms used in the teams.

5.6.1. Extending the definition

Genres may be considered at different levels of abstraction and they can be combined in *genre repertoire*, “a set of genres routinely enacted by a particular community” [20]. A community’s genre repertoire indicates “its established communicative practices and it can serve as a analytic tool for investigating the establishment of a community’s communicative practices” [20]. The concept of repertoire can also be used in relation with the coordination mechanisms, as teams can establish a repertoire of coordination mechanisms, that indicate the coordinative practices enacted in the team. In [18], the authors referred to the concept of ecology of channels, comprising both the concepts of repertoire of communicative genres and the concepts of repertoire of coordination mechanisms. However, a distinction between the two is useful for understanding how the repertoire develops during the project, how it gets established and what are the relations among the two in a project. Some examples are reported in the following.

5.6.2. Examples

The student projects have been studied during the whole collaboration and full access to all artifacts produced and tools used has been provided to the researchers; thus, it is possible to provide an overview of the repertoires developed in the teams, as it is shown in the first two examples reported in the following. The third example describes the role of IM as a dispatcher in the repertoire used by the DHI team, as team members are aware of the repertoire of tools they use and they share social protocols on how to use them.

Repertoire of Coordination Mechanisms. In [32], the authors described in detail the establishment and development of the repertoire of communicative genres in Team C of the student project case and they relate it with the establishment of the repertoire of coordination mechanisms. Five coordination mechanisms are established in the team: the issue managing mechanisms, the versioning of the source code, the file sharing mechanism, the standup meeting and the sharing of the agenda for the meetings. All coordination mechanisms are supported by a functionality available in Assembla. However, while the first three mechanisms are supported by tools traditionally used in Software Engineering, such as the issue tracker tool, the version control system and the file sharing system, the last two coordination mechanisms - the standup meeting and the agenda - are supported by the Forum. A detailed description of all coordination mechanisms is provided in the paper, as well as the supportive role of the communicative genres repertoire for the negotiation and establishment of the repertoire of coordination mechanisms [32]. In summary, the empirical results of the paper show that the role of SoSo is to support informal communication, enabling social talks and metawork, both necessary for establishing and for maintaining effective coordination mechanisms, thus successful cooperation [32].

Repertoire of Communicative Genres. For the purpose of this paper, a detailed analysis of the whole communication through Forum messages in the three student teams has been performed through the communicative genre analytic tool. Five communicative genres have been identified: team building, situated articulation, metawork, work and knowledge sharing. A description of each genre is provided in [32] and most of the genres have also been explained in the previous subsections. Once SoSo logs have been coded through genre analysis, the numerical distribution of genres can provide some insights about the teams, as reported in Figure 4, that indicates which communicative genres are developed more often by each of the three student teams. Please note that in our field material, one Forum message is coded with one single genre. If more genres appear to be suitable for a message, the main genre is assigned. Figure 4 gives some indications about the amount of communication occurred in the different teams, in particular highlighting that Team C invested a lot in team building and articulation work discussions, while in Team A metawork and team building genres did not occur as often. However, analyzing only the repertoire of communicative genres does not provide insights on the collaboration. By relating the communication repertoire with the repertoire of coordination mechanisms, it is possible to provide a description of practices - as the authors did in [32] - and hypothesis for the reasons of possible breakdowns - as discussed in Section 6.4.

The dispatcher role of IM chats. In the DHI case, a detailed analysis of the repertoire of coordination mechanisms and of communicative genres has not been performed. However, already in [18] the authors referred to the concept of “ecology of channels”, to indicate a socio-technical communication system where different channels are used in a complementary way [18]. Team members use the repertoire and are aware of how to use the different tools, thus they share social protocols on how to use them. Our analysis supports

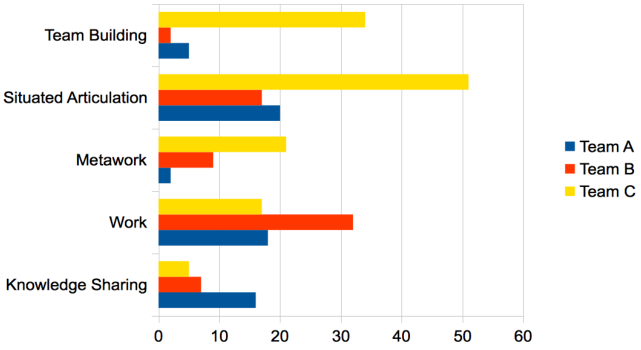


Figure 4: Student Example

the fact that in distributed settings it is important not only to use the same set of tools, but also to develop common social protocols through articulation work and metawork. The analysis provides an indication that IM plays a special role in such socio-technical systeem, acting as a glue between different communicative channels and complementing collaborative SE tools that provide templates for coordination mechanisms. These aspects will be further discussed in Section 8.3.

6. Relation between Communicative Genres and Coordination Mechanisms

The mutually supportive relation between communicative genres and coordination mechanisms is visible in many of the examples reported in Section 5 and it has been already highlighted in several cases. However, the relation among the two concepts is further detailed in this section, through more complex examples, in order to explicate how the theoretical framework presented in Section 7 comes about.

6.1. Example 1: File Notifier genre and File Sharing mechanism

In the coordination mechanism repertoire of the students teams, described in Section 5.6, the *file sharing coordination mechanism* is reported as part of the coordinative repertoire established in the team. In the file sharing system available in Assembla, important files (*artifacts*) are regularly exchanged and, when the file is considered of particular relevance, a message is posted in the Forum to notify to other team members that the file is available in the system (*social protocol*). The Forum message constitutes a *File Notifier genre*, that has been described in Section 5.4, as part of situated articulation genre category occurring in the student teams, and that in this case supports the coordination mechanism through explicit communication mediated by SoSo. Whenever coordination mechanisms are established, situated articulation genre in SoSo sustains the coordination mechanism, allowing team members to keep an informal channel open for further discussions or for social talk. Another example of this kind - situated articulation genre that enacts and enhances the coordination mechanism - is visible in the DHI case and it is reported in Section 5.5: situated articulation chats containing incident numbers are exchanged among team members and sustain the established coordination mechanism of issue tracking supported by Spira.

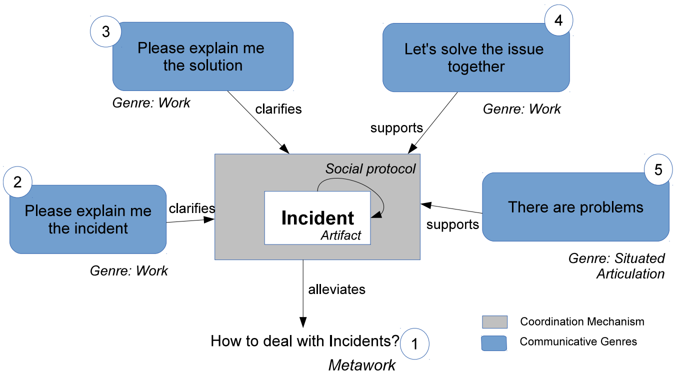


Figure 5: The Incident Coordination Mechanism is supported by Work Genre and Situated Articulation Genre.

6.2. Example 2: DHI incident

In Section 5.1, the *Incident Coordination Mechanism* established in the DHI team is described. The process described in the incident workflow available in the documentation is supported by communication - through mails, voice calls, screen sharing and IM chats - that takes place in order to clarify the formal specification and that adds information to the descriptions reported in Spira: different communicative genres support the Incident coordination mechanism. Following the lifecycle of a specific incident, we identified three phases in which communication takes place across sites [18]: (a) when the defect is assigned to the PAM, (b) when the developer works on the defect, and (c) when the defect is in status “completed” - see Figure 1 for the representation of the incident workflow.

Figure 5 shows a representation of the *Incident coordination mechanism* and of the communicative genres that support it. The coordination mechanism alleviates the necessity of ad-hoc discussions on how to deal with incidents when they are found in the code (1). In phase (a), if a tester or a developer assigns the defect to a PAM, the steps to reproduce the incident described in Spira could be insufficient for the PAM to reproduce it. In this case, the PAM starts an IM chat with the Indian member through IM (*work genre*), or he may ask to start a screen sharing session in order to better understand the problem (2). When a developer is working on the resolution of an incident (phase b), and he has doubts on how to implement the solution described in Spira, he contacts the PAM to discuss the technical issues (*work genre*), through IM or audio (3). Finally, when an incident is closed (c), it can happen that the implemented solution does not completely solve the issue, or the solution can interfere with other parts of the software. In the former case, the PAM uses chats and audio to talk directly with the developer and solve the issue (*work genre*) (4) or to decide to re-open the incident (*situated articulation genre*) (5). In the latter case, a mail is sent to all the PAMs and developers involved, to inform all members about the problem and to find a shared solution (*situated articulation genre*) (5). The resolution of incidents

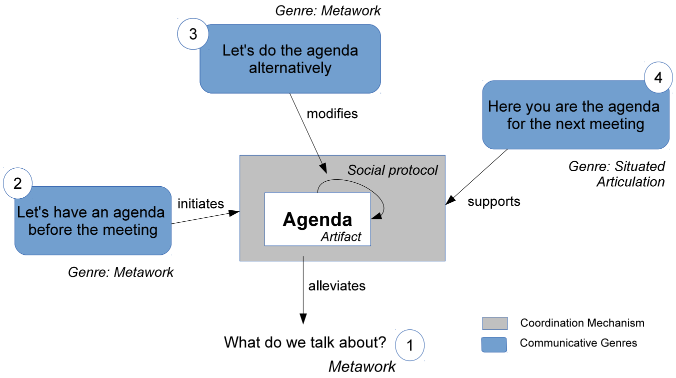


Figure 6: The Agenda Coordination Mechanism is supported by Metawork Genre and Situated Articulation Genre.

takes place not only through the coordination mechanism enacted by Spira and by the source code repository, but it involves a whole range of communicative genres supported by different communication channels - mail, chat and screen sharing - that sustain the coordination mechanism, allowing the success of the collaboration. Among the different channels used, IM chats are the initial dispatcher through which other channels are then considered and used, if needed.

6.3. Example 3: student agenda

Section 5.1 desribes how the *agenda coordination mechanism* is established in Team C of the student case. Figure 6 shows a more complete representation of how the coordination mechanism is established, highlighting the relationship between communicative genres and coordination mechanisms. The agenda is a coordination mechanism that alleviates the articulation work needed to decide what should be discussed during a meeting (1). In reaction to a not successful meeting, the team discusses the usage of agenda in a Forum thread and decides to share the agenda before the meeting, thus *metawork genre* serves to initiate the coordination mechanism (2). Once the coordination mechanism is defined and adopted, a rediscussion of the protocol occurred through Forum messages, deciding that the agenda should be produced by both sub-teams alternatively each week; thus the social protocol evolves over time thanks to *metawork* (3). Once team members have decided who is responsible to share the agenda before the meeting, the sub-team responsible creates a Forum message that is broadcast to all team members and that communicates the agenda of the week; thus, *situated articulation genre* takes place in Forum as part of employing the coordination mechanism, supporting and enhancing it (4). The coordination mechanism is initiated, modified and enacted through two different communicative genres - metawork and situated articulation -, both supported by Forum messages.

6.4. Example 4: comparing three student projects

Section 5.6 reports the analysis of the repertoire of communicative genres appearing in the Forum messages of the student teams case, allowing to compare the three teams. However, it is necessary to relate the repertoire of communicative genres with the repertoire of coordination mechanisms established in each team in order to describe the different teams and to show possible reasons for breakdowns. A detailed analysis of the communicative and cooperative practices in the three student teams is reported in the following and a synthesis is provided at the end of this subsection to provide insights about the collaboration.

Team A. Figure 4 shows that team members scarcely communicated through Forum, mostly using it for situated articulation and rarely for metawork or team building. Most of the conversations are pushed from the Danish team and seldom Brazilian students use Forum, mails or IM chats to communicate with the remote team. The team mostly relied on weekly meetings and on some coordination mechanisms. In particular, the team effectively established one coordination mechanism: the sharing of the agenda, supported every week, by a situated articulation message that was posted in the Forum alternatively by each sub-team. Many of the other coordination mechanisms do not seem to be used in a common way across sites. For example, the file sharing coordination mechanism was used by both teams but the social protocol was not shared among team members: the File notifier genre through Forum messages was used solely by the Danish sub-team. An example of coordination mechanism that has been initiated, but not adopted, is the issue managing system, that was initiated by the Danish team, creating 50 issues throughout the project; however, issues have never been really used by the Danish team, nor by the Brazilian team. The lack of communication and of shared social protocols caused misunderstandings and major challenges in the last part of the project, when the two parts of the code developed independently by the two remote teams needed to be integrated. An example of issues related with the lack of articulation work is reported in a metawork Forum message:

> If anybody else engage in testing of the remaining components, then it would be nice if you would post a message so we don't do double work and create conflicts in SVN as Roberto and I did earlier today :-) *[Jonas, Denmark]*

The collaboration of Team A is proven to be not successful - as visible in the field material and as reported by team members in the final report - and many reasons could be hypothesized to explain it. However, the analysis of communicative and coordinative practices can provide some indications about the misalignment between communicative genres and coordination mechanisms. Metawork did not take place in any tracked communication tool - it occurred only during weekly meetings rather than in mail or SoSo - and, coordination mechanisms did not effectively get established among team members. Metawork can certainly happen during meetings, however, in this case, it did not appear sufficient to get a shared understanding of the negotiated practices to follow by sub-teams during the project. Moreover, the team did not invest in team building through SoSo, that in other teams - e.g. DHI case and Team C -

seems to encourage and sustain the collaboration. To conclude, from the analysis of Team A, we can affirm that relying only on pre-established coordination mechanisms does not appear advisable, especially if social protocols on the usage of the coordination mechanisms are not successfully negotiated and shared among team members.

Team B. The collaboration started with many challenges and team members tried to overcome them imposing processes and social protocols to improve the collaboration through many metawork proposals - see Figure 4. However, negotiation and shared adoption of practices did not occur and imposed processes were not followed by all team members. For example, daily stand up meetings were proposed to give to the project an everyday life time; however, they have been used rarely since - as stated in the final report of the Danish team - "they appeared to be a quite formal requirement". Team building was not invested by team members and was not intertwined with the metawork and with the actual work, as it is visible in Team C; this can be seen as one of the factors influencing the not successful establishment of common practices. As the project progressed, though, different social protocols got established from what was explicitly decided in the beginning, and the initial metawork has been replaced by practices derived from the actual work. Some explicit coordination mechanisms got established in the last part of the project. For example, the issues managing system has been used only in the last month of the collaboration (22 issues in total), despite software development and collaboration activities were already taking place in the previous months. The team often used the File Notifier genre - reported as part of situated articulation in Figure 4 - to highlight the uploading of some important files; however, no regular patterns are visible, as reported instead for Team C. As Figure 4 shows, most of the messages exchanged in Team B are work discussions. Since both teams have responsibilities for all parts of the software, many dependencies in the actual collaboration arose and teams needed many work discussions, especially in the last part of the project, the Forum was effectively used to support them and its asynchronous nature helped to overcome the lack of overlapping working hours. However, the big amount of work discussion can also be an indication for the scarce effectiveness of the coordination mechanisms adopted. In summary, as for Team A, in Team B metawork did not turned out to be successful; moreover, the team did not use SoSo to push team building conversations, thus to establish a good environment for the collaboration. Nevertheless, the team finally succeeded in the collaboration, despite encountering difficulties in the integration of the software developed by the two sub-teams, that required intense collaboration in the last part of the project.

Team C. Figure 4 shows that much more communication through Forum is visible in Team C than in other teams. In particular, the team had many team building, metawork and situated articulation chats, that had a positive influence on the actual collaborative work performed. The work discussions were mainly performed during weekly meeting, in the weekly standup and in the group chat. In the team, socialization and encouraging chats are mixed with the actual work. Team members made effective usage of many coordination mechanisms: 110 issues have been produced and used as the main coordination mechanism for the software development activities. The large amount of File Notifier messages - reported as part of situated articulation in Figure 4 - demonstrates that also the sharing of minutes, of the agenda and of the standup meeting worked as effective coordination mechanisms and were adequately supported by the situated articulation genre. For example, the minutes notifier genre supported the coordination mechanism of sharing the minutes of the meetings; producing and sharing the minutes of the meeting was an initiative of the Danish team and was not a practice commonly defined or requested; however, it became an established practice. Chinese members considered it very important, since it helped them "to go through the discussions that occurred in the meeting and check if there was a common understanding of what was discussed". It is interesting to note that, in Team C, practices got established after the first month of collaboration and patterns of collaboration are clearly visible in the Forum, e.g. as sequences of messages: standup, agenda, minutes -. Most of the situated articulation messages effectively supported the different coordination mechanisms established. The development of the repertoires in Team C is described in detail in [32], however, from the description provided, it is evident the supportive role of the Forum for initiate, re-negotiate - through metawork - and sustain - though situated articulation - the coordination mechanisms. The high amount of team building seems to have an impact on the cohesion of the team that supported the collaboration.

In summary. The communicative and coordinative practices of the three student teams have been described in detail in the previous paragraphs using the concepts of communicative genres and coordination mechanisms in a complementary way. Thanks to the analysis performed, it is possible to hypothesize possible reasons for the success of the teams or for the breakdowns.

Team A relied solely on professional coordination mechanisms, without dedicating time and effort on metawork - for the negotiation and establishment of shared social protocols -, nor on team building - for encouraging a good work environment -, thus the team experienced major challenges in the final phase of the project. Also in Team B initial metawork did not turned out to be successful and no team building occurred. The team finally succeeded in the collaboration, though encountering difficulties in the integration of the software developed by the two sub-teams, that required intense collaboration in the last part of the project. Whereas, Team C succeeded with the establishment of shared coordination mechanisms and effective social protocols thanks to the initial successful metawork and to team building. In Team C, metawork and situated articulation have been fundamental to establish, maintain, and enact coordination mechanisms, that allowed to establish a smooth collaboration and a positive working environment.

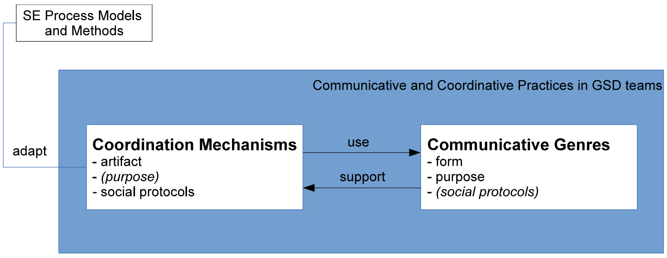


Figure 7: Conceptual Framework

Thus, the hypothesis is that the development of satisfying communication and coordination practices depends on the successful establishment and maintaining of social protocols, which seems to be dependent on the initial metawork and on team building. However, these hypotheses that arise from the analysis performed need to be confirmed through further investigations. These aspects will be re-discussed in Section 8.

7. The Conceptual Framework

This section presents the conceptual framework that we propose for analyzing and describing coordinative and communicative practices in GSD, relating the notions of coordination mechanisms and communicative genres, described and extended in the previous section. Figure 7 shows the framework: existing software process models and methods need to be adapted and appropriated by team members in situated action [26]. The development of common social protocols is crucial for the development of effective communicative and coordinative practices and for the adaptation of models and methods. The concepts of coordination mechanisms and communicative genres are based on the notion of social protocols and are mutually supportive, thus they allow to describe how communicative and coordinative practices are developed and maintained in GSD teams. Coordination mechanisms provide a way to collaborate through digital artifacts, alleviating articulation work. However, they are supported by explicit communication, thus by communicative genres. Thanks to communicative genres, coordination mechanisms can be initiated, discussed, established and maintained, thus they can support the collaboration among remote team members. Section 6 reports several examples to describe communicative and coordinative practices adopted in different GSD teams - both novel and established ones - using the theoretical concepts of this framework and showing the mutually supportive relationship between communicative genres and coordination mechanisms. Implications for research and practice of the proposed framework are discussed in the following section.

8. Discussions

This section presents and discusses the main contributions of this article. Initially, the usefulness of the conceptual framework is established and the importance of repertoires is highlighted, then the role of Social Software (SoSo) in the repertoire is discussed. Finally, implications for research and for practice are reported and limitations are presented together with future research directions that arise from the findings of this paper.

8.1. On the Conceptual Framework

This paper presents a novel conceptual framework that integrates and extends the analytic tools of coordination mechanisms and communicative genres, showing how they mutually support each other. On the one hand, coordination mechanisms alleviate articulation work; on the other hand, communicative genres with the purpose of articulation work can initiate, redefine, enhance and support coordination mechanisms, alleviating the necessity of further and more complex articulation work. Using the framework, we can in detail see when and how successful metawork takes place and results in commonly adopted coordination mechanism, allowing to understand how and why metawork and coordination mechanisms fail. The theoretical framework is constructed upon the empirical material of the cases described, however it can be used to describe also practices in other GSD projects. For example, a study by Damian et al. [44] shows that coordination mechanisms can not work due to possible misunderstandings; this paper provides a language to express why it happens, showing how the mismatch comes about and what would be needed to address it. Using our theoretical framework, it is possible to explain that, in Damian et al. [44] case, Canadian team relied on coordination mechanisms, while the US team was expecting situated articulation messages through mail in order to be informed on the changes to the code. Thus, in the case described, the metawork was not successful, as it did not allowed to establish shared coordination mechanisms

As another example, the conceptual framework allows to describe the media and the subject components of the software informalisms proposed by Scacchi [23]. The media side of the software informalism can be related to the communicative genre, while the subject is supported by coordination mechanisms. Therefore, e.g. a Forum message that contains the agenda of a meeting can be considered as a situated articulation genre - the media - as well as as the artifact of the coordination mechanism - the subject. Scacchi considers Forum threads as software informalisms; thanks to our conceptual framework it is possible to describe with more detail how team members use the artifacts and how software informalisms are developed and adopted in GSD teams. In summary, the conceptual framework allows to better understand existing research and provide researchers a language to describe practices and to express breakdowns. Moreover, the conceptual framework allows to explain that in GSD it is not sufficient to use the same set of tools in order to effectively collaborate remotely: tools need to be adapted to team's necessities and processes and models need to be adopted by team members. This is further detailed in the next subsection.

8.2. About The Repertoire

The conceptual framework offers a way to describe whether and how team members succeed in adaptation and adoption of methods and tools and it shows how practices evolve during time, through the evolution of social protocols. Coordination mechanisms and communicative genres are part of a repertoire (or ecology of channels [18]) that can be - and has to be - studied as a unitary system in order to understand the relationships between different channels and tools. The examples reported in this paper show that not one specific set of coordination mechanisms and communicative genres is recommended, as different repertoires are adopted in different teams. However, the framework suggests a set of function has to be supported by the repertoire in order to achieve successful collaboration, such as - but not exclusively - metawork and socialization. Future research could investigate in detail what are the additional needs that have to be part of a repertoire to make it effectively.

8.3. Understanding the Role of Social Software

A literature review performed by the authors [12] highlights the lack of research papers focusing on understanding the relationship between SoSo and the ecology of tools used by GSD teams. This topic is addressed in this research work. SoSo is part of an ecology of channels [18] that has to be explored as a whole, not exclusively focusing on the specific functionality of each kind of SoSo. Thanks to the theoretical framework, the central role of SoSo in the teams analyzed is highlighted: on the one hand, it supports metawork to enact and negotiate coordination mechanisms; on the other hand, it supports different kinds of communicative genres, e.g. work discussions, knowledge sharing, articulation work and team building. SoSo complements Software Engineering collaborative tools, that function as coordination mechanisms, supporting communication among team members. The analysis through the theoretical framework of different GSD teams reported in this paper shows the dynamics of how social protocols are negotiated and established within novel GSD teams, it provides an indication why breakdowns occur, and it describes how social protocols are maintained in an established team, highlighting the supportive role played by SoSo.

SoSo is fundamental in the establishing phase of projects, when decisions need to be taken, social relationships need to be established and social protocols need to be negotiated. After the initial phase, Forum and Wiki serve as a persistent repository for the knowledge shared, while the communication through Forum and IM serve as channel for situated articulation, decision making and collaboration record. IM is the media where things happen if they can not take place somewhere else, acting as a dispatcher for other channels and as the main channel for social talk. SoSo enables social talks and metawork, both necessary for establishing and maintaining successful collaboration. On the one hand, social talk support the actual work in distributed environments by enhancing the development of social relationships among team members. On the other hand, metawork is fundamental for the software development activities, allowing to negotiate and establish social protocols in the collaboration. Providing to distributed teams the access to flexible tools such as SoSo, and encouraging team building chats appear to have an impact on the success of the collaboration. Not allowing for social talk may turn out to be expensive.

8.4. Implications for Research and for Practice

The present paper shows how situated action [26] can be analyzed to study communicative and coordinative practices in GSD teams. Practices in situated action are not fully specified by software engineering methods and processes. Thus, adaptation by the teams is necessary. The concept of social protocols as part of both communicative genres and coordination mechanisms helps to further explore these processes. A theoretical framework based on the notion of social protocols is useful to understand breakdowns, to investigate the establishment of practices in novel GSD teams and to analyse the re-negotiations of practices in established teams. The importance of negotiating and agreeing on common social protocols is particularly decisive in GSD settings, in which direct communication can not take place so easily - as it is often mediated by artifacts - and in which socio-cultural distance can affect the collaboration. Thus, agreeing on social protocols becomes even more challenging and crucial. The theoretical framework can be beneficial for future research that aims to analyze and describe not only the role of SoSo, but also how communicative and coordinative practices are established and maintained in GSD teams.

An encouraging contribution, both for researchers and for practitioners, is that the success of the collaboration strongly relates on how coordination mechanisms and social protocols are established among team members, helping to bridge language barrier, cultural differences, time and space distance. The analysis indicates the importance of communication supported by SoSo as a side channel that complements professional coordination mechanisms. It does not appear necessary to design new tools, while it is very important to understand how existing tools are adopted and what are the potentials of their usage. Other channels than SoSo can certainly be used, however it seems advisable for practitioners to make sure that metawork and social talk are supported by tools used by team members and that they somehow take place, as geographical distribution does not preclude the possibility to have these kinds of conversations.

8.5. Limitations and Future Works

The theoretical framework allows to see the heterogeneity of channels and artifacts used in the teams. However, the analysis does not aim to be exhaustive: further genres can be identified and genres categories can occur in different media than the ones analyzed in the present paper. For example, virtual meetings could be likewise analyzed, adapting the communicative genres analytic tool. Moreover, the coordination mechanisms described in this article are mostly established and persistent during the collaboration; however, it could be possible to identify further temporary coordination mechanisms established by team members for specific purposes driven by particular situations occurring during the remote collaboration. Thus, further communicative and coordinative practices can be included while studying further cases through the framework.

Another limitation is that the theoretical framework focuses on communicative and coordinative practices: common practices adopted by GSD teams definitely include also other kinds of practices, such as coding practices or design practices. We kept the focus on on communicative and coordinative practices, as they allow to understand the role of SoSo in the cooperation across distance: thanks to the framework proposed, it is possible to investigate and understand the role of SoSo within the repertoire used in the projects studied. However, future research could extend the framework including additional conceptual tools for analyzing and describing further kinds of practices. ‘

9. Conclusions

This paper presents a novel framework to analyze and describe coordinative and communicative practices in Global Software Development (GSD), that is based on the concepts of communicative genres and coordination mechanisms. The conceptual framework offers to researchers a tool to look at computer-mediated collaborative practices in distributed settings highlighting the importance of social protocols. It allows to better understand existing research, the heterogeneity of practices in GSD teams and the role of Social Software (SoSo) within the repertoire used in GSD teams. Through the framework, it is possible to highlight that SoSo allows team members to establish, develop and maintain social protocols during the collaboration and to support metawork and team building chats. The theoretical framework can be beneficial for future research that aims to analyze and describe not only the role of SoSo, but also how cooperative practices are established and maintained in GSD teams.

10. Acknowledgments

The authors would like to thank Dr. Irina Shklovski for the insightful comments and feedback on on an earlier version of this article. Moreover, the authors would like to thank Paolo Tell for the support with the coding of the field material, for indispensable discussions on the topic and for the always-on research support.

11. References

[1] Y. Dittrich, D. W. Randall, J. Singer, Software engineering as cooperative work, Computer Supported Cooperative Work (CSCW) 18 (5) (2009) 393–399.

[2] K. Schmidt, L. Bannon, Taking cscw seriously, Computer Supported Cooperative Work (CSCW) 1 (1-2) (1992) 7–40.

[3] K. Schmidt, C. Simone, Coordination mechanisms: Towards a conceptual foundation of cscw systems design, Computer Supported Cooperative Work (CSCW) 5 (2-3) (1996) 155–200.

[4] J. D. Herbsleb, Global software engineering: The future of socio-technical coordination, in: 2007 Future of Software Engineering, IEEE Computer Society, 2007, pp. 188–198.

[5] E. Carmel, R. Agarwal, Tactical approaches for alleviating distance in global software development, Software, IEEE 18 (2) (2001) 22–29.

[6] J. D. Herbsleb, A. Mockus, An empirical study of speed and communication in globally distributed software development, Software Engineering, IEEE Transactions on 29 (6) (2003) 481–494.

[7] K. Ehrlich, M. Cataldo, All-for-one and one-for-all?: a multi-level analysis of communication patterns and individual performance in geographically distributed software development, in: Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work, ACM, 2012, pp. 945–954.

[8] T. Niinimaki, C. Lassenius, Experiences of instant messaging in global software development projects: A multiple case study, in: Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on, 2008, pp. 55 –64. doi:10.1109/ICGSE.2008.27.

[9] T. LaToza, G. Venolia, R. DeLine, Maintaining mental models: a study of developer work habits, in: ICSE '06: Proceedings of the 28th international conference on Software engineering, 2006.
URL http://portal.acm.org/citation.cfm?id=1134285.1134355

[10] E. Knauss, O. Brill, I. Kitzmann, T. Flohr, SmartWiki: Support for high-quality requirements engineering in a collaborative setting, in: Wikis for Software Engineering, 2009. WIKIS4SE '09. ICSE Workshop on DOI - 10.1109/WIKIS4SE.2009.5069994, 2009, pp. 25–35.
URL 10.1109/WIKIS4SE.2009.5069994

[11] J. Zhang, Y. Qu, J. Cody, Y. Wu, A case study of micro-blogging in the enterprise: use, value, and related issues, in: CHI '10: Proceedings of the 28th international conference on Human factors in computing systems, 2010.
URL http://portal.acm.org/citation.cfm?id=1753326.1753346

[12] R. Giuffrida, Y. Dittrich, Empirical studies on the use of social software in global software development-a systematic mapping study, Information and Software Technology.

[13] J. D. Herbsleb, D. Moitra, Global software development, Software, IEEE 18 (2) (2001) 16–20.

[14] M. Cataldo, M. Bass, J. D. Herbsleb, L. Bass, On coordination mechanisms in global software development, in: Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on, IEEE, 2007, pp. 71–80.

[15] H. Holmström, B. Fitzgerald, P. J. Ågerfalk, E. Ó. Conchúir, Agile practices reduce distance in global software development, Information Systems Management 23 (3) (2006) 7–18.

[16] L. Layman, L. Williams, D. Damian, H. Bures, Essential communication practices for extreme programming in a global software development team, Information and software technology 48 (9) (2006) 781–794.

[17] F. Lanubile, Collaboration in distributed software development, in: Software Engineering, 2009.

[18] Y. Dittrich, R. Giuffrida, Exploring the role of instant messaging in a global software development project, in: Global Software Engineering (ICGSE), 2011 6th IEEE International Conference on, IEEE, 2011, pp. 103–112.

[19] S. Black, J. Jacobs, Using web 2.0 to improve software quality, in: Proceedings of the 1st Workshop on Web 2.0 for Software Engineering, ACM, 2010, pp. 6–11.

[20] W. J. Orlikowski, J. Yates, Genre repertoire: The structuring of communicative practices in organizations, Administrative science quarterly (1994) 541–574.

[21] E. Ó. Conchúir, P. J. Ågerfalk, H. H. Olsson, B. Fitzgerald, Global software development: where are the benefits?, Communications of the ACM 52 (8) (2009) 127–131.

[22] Y. Wang, E. Trainer, B. Al-Ani, S. Marczak, D. Redmiles, Attitude and usage of collaboration tools in gse: A practitioner oriented theory, in: Cooperative and Human Aspects of Software Engineering (CHASE), 2012 5th International Workshop on, IEEE, 2012, pp. 135–137.

[23] W. Scacchi, Free/open source software development: Recent research results and methods, Advances in Computers 69 (2007) 243–295.

[24] I. R. McChesney, S. Gallagher, Communication and co-ordination practices in software engineering projects, Information and Software Technology 46 (7) (2004) 473–489.

[25] T. W. Malone, K. Crowston, The interdisciplinary study of coordination, Computing Surveys 26 (1) (1994) 87–119. doi:10.1145/174666.174668.

[26] L. A. Suchman, Plans and situated actions: the problem of human-machine communication, Cambridge university press, 1987.

[27] K. Crowston, H. Annabi, J. Howison, C. Masango, Effective work practices for software engineering: free/libre open source software development, in: Proceedings of the 2004 ACM workshop on Interdisciplinary software engineering research, ACM, 2004, pp. 18–26.

[28] J. Whitehead, Collaboration in software engineering: A roadmap, in: Future of Software Engineering, 2007. FOSE'07, IEEE, 2007, pp. 214–225.

[29] D. E. Strode, S. L. Huff, B. Hope, S. Link, Coordination in co-located

agile software development projects, Journal of Systems and Software 85 (6) (2012) 1222–1238.

[30] B. Jordan, A. Henderson, Interaction analysis: Foundations and practice, The journal of the learning sciences 4 (1) (1995) 39–103.

[31] C. Robson, Real world research: a resource for social scientists and practitioner-researchers, Vol. 2, Blackwell Oxford, 2002.

[32] R. Giuffrida, Y. Dittrich, How social software supports cooperative practices in a globally distributed software project, in: Cooperative and Human Aspects of Software Engineering (CHASE), 2014 7th International Workshop on, ACM, 2014.

[33] D. Chandler, An introduction to genre theory, The Media and Communications Studies Site.

[34] J. Yates, W. J. Orlikowski, Genres of organizational communication: A structurational approach to studying communication and media, Academy of management review 17 (2) (1992) 299–326.

[35] C. R. Miller, Genre as social action, Quarterly journal of speech 70 (2) (1984) 151–167.

[36] H.-G. Im, J. Yates, W. Orlikowski, Temporal coordination through communication: using genres in a virtual start-up organization, Information Technology & People 18 (2) (2005) 89–119.

[37] Y. Wang, D. Redmiles, Understanding cheap talk and the emergence of trust in global software engineering: An evolutionary game theory perspective.

[38] E. M. Gerson, Reach, bracket, and the limits of rationalized coordination: Some challenges for cscw, in: Resources, Co-Evolution and Artifacts, Springer, 2008, pp. 193–220.

[39] A. Strauss, The articulation of project work: An organizational process, The Sociological Quarterly 29 (2) (1988) 163–178.

[40] L. Pries-Heje, Four different paradigms for process design when implementing standard enterprise systems, CONFENIS 2010.

[41] L. Wittgenstein, Philosophical investigations, Oxford: Blackwell, 1953.

[42] M. Yasuoka, Bridging and breakdowns, Ph.D. thesis, Copenhagen Business School (2009).

[43] S. R. Barley, The alignment of technology and structure through roles and networks., Administrative science quarterly 35 (1).

[44] D. Damian, L. Izquierdo, J. Singer, I. Kwan, Awareness in the wild: Why communication breakdowns occur, in: Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on, IEEE, 2007, pp. 81–90.