

A Systematic Review of Knowledge Sharing Challenges and Practices in Global Software Development

Mansoorreh Zahedi ^a, Mojtaba Shahin ^b, Muhammad Ali Babar ^{a,b}

^a IT University of Copenhagen, Denmark

^b The University of Adelaide, Australia

Abstract

Context: Global Software Development (GSD) presents significant challenges to share and understand knowledge required for developing software. Organizations are expected to implement appropriate practices to address knowledge-sharing challenges in GSD. With the growing literature on GSD and its widespread adoption, it is important to build a body of knowledge to support future research and effective knowledge sharing practices.

Objective: We aimed at systematically identifying and synthesizing knowledge sharing challenges and practices. We also intended to classify the recurrent challenges and most frequently reported practices in different contextual settings.

Method: We used Systematic Literature Review (SLR) for reviewing 61 primary studies that were selected after searching the GSD literature published over the last 14 years (2000 – September 2014). We applied thematic analysis method for analysing the data extracted from the reviewed primary studies.

Results: Our findings revealed that knowledge sharing challenges and practices in GSD could be classified in 6 main themes: management, team structure, work processes/ practices, team cognition, social attributes and technology. In regard to contextual settings, we found empirical studies were mainly conducted in an offshore outsourcing collaboration model distributed between two sites. Most of the studied organizations were large enterprises. Many of the studies did not report any information for several contextual attributes that made it difficult to analyse the reported challenges and practices with respect to their respective contexts.

Conclusion: We can conclude: a) there is a higher tendency among researchers to report practices than challenges of knowledge sharing in GSD. b) Given our analysis, most of the reported knowledge sharing challenges and practices fall under the theme of “work practices”. c) The technology related knowledge-sharing challenges are the least reported; we discussed the available technologies for supporting knowledge sharing needs in GSD. d) The organizational contextual information is missing from a large number of studies; hence, it was not possible to investigate the potential relations between knowledge sharing challenges/practices and the contextual attributes of GSD teams. We assert the need of exploring knowledge sharing in the context of small/ medium sized organizations to avoid the risk of findings being biased by specific empirical setting (e.g., large enterprises distributed between US and India).

Keywords: Knowledge Sharing, Knowledge Management (KM), Global Software Development (GSD), Systematic Literature Review (SLR), Empirical Software Engineering

1. Introduction

Global Software Development (GSD) has become an established software development paradigm that promises several advantages but also suffers from well-known limitations [1-3]. The promised benefits include enabling organizations to implement strategies like Follow The Sun (FTS), benefiting from cost advantages in certain parts of the world, being in close proximity to customers, and creating opportunities for merger and acquisition, and accessing a large pool of talented software developers [2, 4, 5]. GSD can also result in significantly increased complexity for project teams, who may have to face several kinds of new challenges. Most of the GSD challenges stem from what is known as GSD distances, i.e., temporal, geographical, cultural, and linguistic distances. These distances lead to communication, coordination, and collaboration challenges [5-7] that can impact several areas of software development. One of the key areas of software development being impacted by GSD is knowledge sharing as software development is a

knowledge-intensive activity whose success is largely dependent upon effective knowledge sharing among software development teams [8], [9]. GSD team members may find it difficult (or even impossible) to share both tacit and explicit knowledge within a team that is geographically distributed.

Knowledge sharing is an integral part of Knowledge Management (KM) [10], [11]. It is defined as «*provision of task information and know-how to a person, so that (s) he can collaborate with others to solve problems, develop new ideas or implement policies or procedures*» [11, 12]. Choo and Alvarenga [10] identified four major categories of conditions to enable knowledge sharing [10]: social/ behavioral characteristics of teams (e.g., mutual trust, attentive enquiry, open dialogues), cognitive/ epistemic attributes (e.g., common knowledge, shared values and goals), organizational structure/strategies (e.g. empowered divisions, leadership style) and provision of information systems (e.g., internet, intranet, yellow pages).

Ebert and De Man [13] discuss that effectively managing software engineering knowledge (i.e., project, product, and process) is of growing importance given the ever-changing environments of software development, e.g., globalization. They [13] argued that in GSD settings, vendors usually possess technical expertise and knowledge about a project, while clients hold requirements and application domain knowledge. However, vendors without proper understanding of requirements and application domain knowledge cannot effectively and efficiently apply their technical skills; and clients could not appropriately provide business requirements without understanding knowledge needs of vendors. Bjornson and Dingsøyr [14] report that KM in software engineering is mainly based on utilizing information technologies.

Boden and colleagues [15] criticize the implementation of traditional KM approaches that tend to de-contextualize by codifying knowledge but neglect behavioral aspects and social learning that are specifically influential in GSD teams. GSD impedes the opportunities for face-to-face interaction and informal chats between distributed team members [5, 6] which are considered quite helpful for sharing knowledge [5, 16]. Time zone differences also decrease the mechanisms of *ad-hoc* knowledge sharing by answering on-the-spot questions. Cultural and linguistic distance plays a significant role in communication gaps between distant colleagues. From socio-behavioural perspectives, several studies (e.g., [8, 17]) indicate that developing trust and rapport between dispersed members facilitate knowledge sharing. The argument is that when individuals are socially bonded they are more likely to share identity [17] and cooperate with each other due to trustworthiness, obligations, and expectations [18]. Yet, building and maintaining a social network among individuals in the absence of face-to-face communication and informal chats could be a costly undertaking.

Given the increasing trend of GSD and the importance of knowledge sharing in GSD, researchers and practitioners have been dedicating significant amounts of effort to help understand knowledge sharing challenges and devise appropriate practices to address the challenges. This has resulted in a growing amount of literature that warrants periodic reviews in GSD. Hence, we decided to systematically identify and critically review the literature on knowledge sharing challenges and practices in GSD.

Our review aims at contributing to a growing body of knowledge on knowledge sharing. We assert that systematic and periodic reviews will help build an evidence-based body of knowledge about knowledge sharing challenges and relevant practices in GSD. Such a body of knowledge can inform the research community about commonly reported (or unaddressed) challenges and direct provision of solutions to support knowledge sharing needs of GSD teams. Besides, it could be used as comprehensive guidelines for practitioners to become more aware of the challenges and implement appropriate practices that suit their work context. That is why we decided to conduct a systematic literature review (SLR) in order to systematically collect, investigate and summarize knowledge sharing challenges and practices of GSD teams from real-world scenarios.

We consider the study by Nidhra et al. [19] as the most relevant to our review. The authors have reported knowledge transfer challenges and mitigation strategies in GSD. Whilst our study and the review by Nidhra and colleagues [19] are on the same topic, both studies have significant differences that we will explain in the later part of this paper.

The rest of this paper is organized as follows: Section 2 describes details of conducting this SLR. Section 3 reports the demographic information of the reviewed primary studies. Section 4 analyzes the research methodologies used by the reviewed studies. Section 5 analyzes the contextual settings of the reviewed studies. Section 6 and 7 report the knowledge sharing challenges and practices identified. The limitations of this study are discussed in section 8. Section 9 compares our study with an existing SLR on this topic. The results are discussed in section 10 and conclusions are provided in section 11.

2. Method

As previously stated, we used a Systematic Literature Review (SLR) that is one of the most widely used research methods of Evidence-Based Software Engineering (EBSE) [20]. SLR provides a well-defined process for identifying, evaluating, and interpreting all available evidence relevant to a particular research question or topic [21]. For this review, we followed Kitchenham and Charters's guidelines [21] for applying a SLR research method that involves three main phases: defining a review protocol, conducting the review, and reporting the review. Our review protocol consisted of these elements: (i) research questions, (ii) search strategy, (iii) inclusion and exclusion criteria, (iv) study selection, (v) study quality assessment, and (vi) data extraction and synthesis. We discuss the followed steps in the following subsections.

2.1. Research Questions

Table 1 presents the research questions (RQs) and their respective motivations. We aimed at gaining an understanding of knowledge sharing challenges (i.e. RQ1), and practices (i.e. RQ2) reported by empirical studies in GSD, and identify the contextual settings from which the challenges and practices are found (i.e. RQ3).

Table 1 – Research Questions of this SLR

Research Question	Motivation
RQ1: What are the knowledge sharing challenges in GSD?	To get an overview of different types of knowledge sharing challenges reported in GSD and identify the challenges that are most frequently reported in the context of GSD.
RQ2: What are the knowledge sharing practices in GSD?	To gain an understanding of the knowledge sharing practices implemented in GSD.
RQ3: In what contextual settings (i.e., research methodology and organizational context) are challenges and practices reported?	To understand the methodological and organizational contextual settings from which challenges and practices have been reported. Such information can help a reader to better understand the reported research.

2.2. Search Strategy

Defining a search strategy for a SLR is considered as one of the most important prerequisites. A search strategy can help the researcher to retrieve as many relevant studies as possible [21, 22]. The search strategy used for this review is as follow:

2.2.1. Search String

We formulated our search string based on three compartments shown in Table 2. We decided not to include a condition for choosing empirical studies in our search string, but excluding non-empirical studies during the review process. Given the variety of research methods, having that condition could have made our search string quite complex. There was also a risk that the studies whose meta-data did not have the information about the used research method might have been unintentionally excluded.

Table 2 – Structure of Search String

A	“Global Software Development” and all synonyms
B	“Knowledge Management” OR “Knowledge Sharing” OR “Knowledge Transfer” with all the synonyms
C	“Challenges” OR “Solutions” with all synonyms

Running a pilot search and verifying the inclusion of papers that we were already aware of, we utilized the final search string as presented in following:

TITLE-ABS-KEY (("global software development" OR "collaborative software development" OR "global software engineering" OR "distributed software development" OR "distributed software engineering" OR "offshore software development" OR "offshore software engineering" OR "geographically distributed software development" OR offshor* OR "software outsourcing" OR "software outsource" OR "globally distributed software development" OR "offshore outsourcing" OR "Dispersed teams" OR "distributed teams" OR "virtual teams" OR "globally distributed work" OR "global software teams" OR outsour*)

AND ("knowledge transfer" OR "knowledge shift" OR "knowledge exchange" OR "knowledge distribution" OR "tacit knowledge" OR "explicit knowledge" OR "knowledge transfer process" OR "knowledge flow" OR "organizational knowledge" OR "knowledge acquisition" OR "knowledge management" OR "knowledge creation" OR "knowledge sharing" OR "knowledge retention" OR "knowledge valuation" OR "knowledge use" OR "knowledge application" OR "knowledge discovery" OR "knowledge integration" OR "knowledge theory" OR "knowledge engineering" OR "experience transfer" OR "technology transfer")

AND (risk* OR challenge* OR tool* OR method* OR problem* OR challeng* OR barrier* OR "best practices" OR model* OR techniq* OR strateg* OR approach* OR process* OR solution* OR obstacle* OR "risk analysis" OR effect* OR "risk factors" OR selection* OR mechanism* OR assessment* OR "evaluation process" OR practice* OR mitigate*))

2.2.2. Data Source

We ran our search string on the Scopus indexing system. This decision was made based on successful experiences of other researchers with using Scopus[23, 24]; that facilitates formulating a single and complex search query [24, 25], frequently updates indices [25] and in particular track a large number of journals and conferences in software engineering and computer science [23, 24] that were the main target of this review. Comparing results of Scopus with a manual search in their study, Kitchenham et al [23] observed that Scopus output covered all the relevant papers that used appropriate terminology and published in mainstream software engineering literature venues.

Furthermore, our pilot search with directly using digital libraries (e.g. Springer Link, Wiley Inter Journal Science, IEEEExplore) revealed a number of restrictions placed by the digital libraries on large-scale searches on the meta-data of the published papers. Moreover, the search string needs to be modified for each single digital library that can result in errors being introduced. Running the search string on Scopus helped us to keep the search string constant while retrieving focused hints that are mostly relevant. Table 3 presents the settings that we used for running the search string. In addition to computer science, we chose all the other subject areas (e.g., business management, decision science, and economics) that may have the potential to publish related studies, especially from the knowledge management perspective.

Table 3 – Details of the Search String Ran on Scopus

Digital Library	Scopus
Years	2000 – 2014
Language	Only English
Run on	Title/ Abstract/ Keyword
Subject Areas	"Computer Science", "Business, Management and Accounting", "Decision Sciences", "Economics, Econometrics and Finance"
Date of running search string	7th September 2014
Number of Hints	1320

2.3. Inclusion and Exclusion Criteria

The inclusion and exclusion criteria are used for selecting relevant primary studies to answer the research questions in a SLR. The inclusion and exclusion criteria were applied to all the retrieved studies from the Scopus database. The criteria used for including/ excluding papers are as follows:

- Peer-reviewed papers only - we excluded editorials, position papers, keynotes, reviews, tutorial summaries, panel discussions and short papers (i.e. less than 6 pages).

- GSD context - Different collaboration models of GSD such as offshore outsourcing, offshore-insourcing, and near-shore are included.
- Studies reporting empirical research using methods such as case studies, survey, experiments, and ethnographical studies.
- The studies based on students' data were included only if they were reporting findings that were expected to contribute to our research questions. If the studies were only from educational perspective (e.g., how to run GSD courses) they were excluded.
- Extended papers - If two papers from the same research on the same topic were published in different venues (e.g., a conference and a journal), only the journal paper was included.
- All duplicated studies found from different sources were detected and removed.

2.4. Study Selection

The number of studies selected at each stage of this SLR is shown in Figure 1. Our search in Scopus returned 1320 results. We filtered the papers by reading the title and abstract. When there were some papers that we could not decide upon by reading the titles and abstracts, these papers were retained for the next round of inspection. We excluded the papers that were shorter than 6 pages, viewed as irrelevant, or whose full text was not available. Since we were interested in empirical studies, we excluded papers that were not supported with empirical data in step 5. Furthermore, we included only those papers that had reported research involving software teams in any type of distributed arrangement. The duplicate papers were also excluded. For example, the shorter versions of studies that were also reported more comprehensively were excluded. At the end of the step 6, we found that 76 papers met all the inclusion criteria.

Having finalized the list of our primary studies, we performed crosscheck with the primary studies included in [19]. Comparing our list with primary studies of the existing SLR [19], we eventually added 8 studies to our list which were either not found by our search or we had excluded them based on title/ abstract reviews. In the eighth step of the papers selection process, 84 papers (i.e., the total papers selected in steps 6 and 7) were distributed between two first authors and they filtered the papers by reading the full text of the papers. In this stage 52 studies were selected as primary studies. Then we followed a snowballing technique [26] to scan the references of these 52 selected papers in order to find more potential primary studies. We found 181 potentially relevant papers by title from the references of these 52 papers. We applied step 2 to step 8 on the 181 relevant papers and finally 9 papers were selected as primary studies. It is worth noting that we excluded the studies published before 2000. The selected studies (i.e., 9 papers) from the snowballing process were included in the final results of the study selection (i.e., 52 papers). That means we selected 61 papers for this review. We recorded the reasons of inclusion or exclusion decision for each of the papers; those reasons were used for further discussion and reassessment about whether or not a paper was to be included.

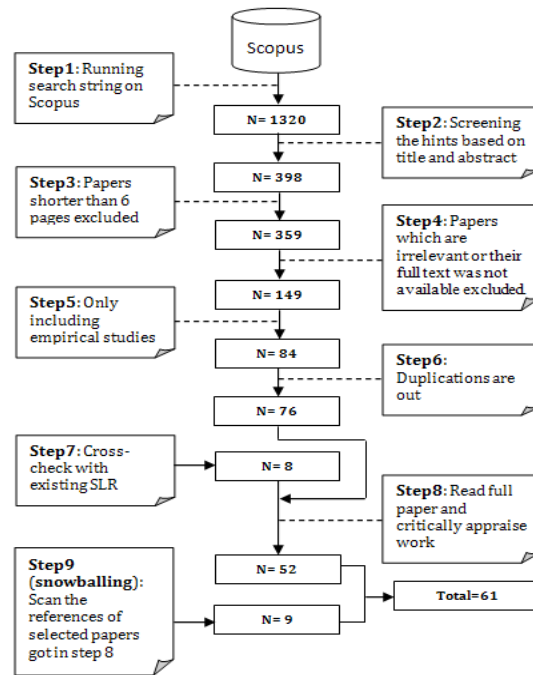


Figure 1 - Steps of the search process and number of selected studies in each step

Appendix A enlists the papers included in this review. We have assigned unique identifiers to the reviewed papers. We refer the papers with their respective unique identifier in this SLR. For example, we use P26 for referring to the paper: “Knowledge Sharing Practices and the Impact of Cultural Factors: Reflections on Two Case Studies of Offshoring in SME” by Boden et al.

2.5. Data Extraction and Synthesis

2.5.1. Data Extraction

The data extraction step of a SLR purports to identify the relevant information that should be extracted from each of the papers in order to answer the research questions. First, the two authors extracted the data using a pre-designed data extraction form and discussed the extracted data with the third author. Appendix B shows the data items for which the relevant information was extracted from the selected papers. It also shows the research question(s) that were supposed to be answered using different pieces of the extracted data. A record of the information extracted from a study was stored in a spreadsheet for subsequent analysis.

2.5.2. Synthesis

For synthesizing the extracted data, we divided the data extraction form into a) demographic and contextual attributes, b) knowledge sharing challenges and practices. The first set of the data items were analysed through quantitative techniques and produced the descriptive results reported in Sections 3, 4 and 5. The second set of data items were analysed using a qualitative analysis method, namely thematic analysis [27]. We transferred the identified challenges and practices into separate Excel sheets and performed analysis phases on each spread sheet independently. We followed the six steps process of thematic analysis developed by Braun and Clarke [27]:

- Familiarizing with data: While initial familiarization was done by reviewing the papers and extracting data, each of the Excel sheets were carefully read through in this stage. It was required to get sense of all the challenges/ practices that were extracted to form the initial ideas for analysis.
- Generating initial codes: the extracted data from each paper was read thoroughly for identifying the reported key points that were assigned appropriate codes. Such coding was possible because the extracted data was elaborative (i.e., challenges/ practices were explained in few sentences), rather than being abstract (i.e., couple of words). We referred back to papers to verify our understanding and getting the context of discussion whenever it was required. This phase was the most time-consuming stage of our analysis, during which we iteratively revised, merged, and defined new codes. Finally, we organized all the data rows based on the assigned codes.

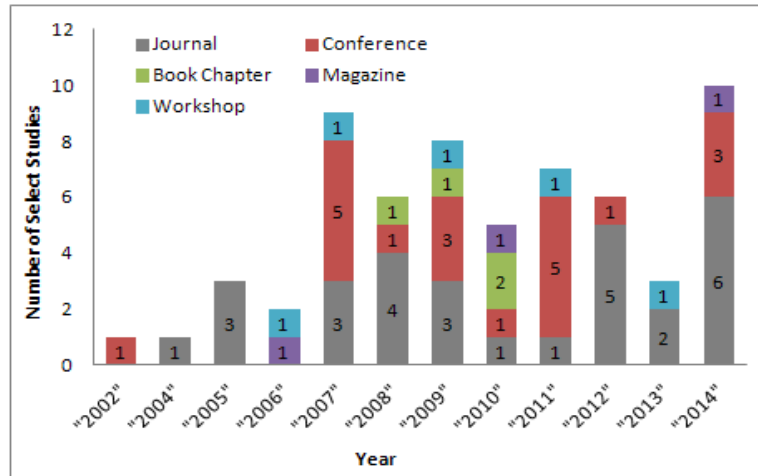


Figure 2 - Number of selected studies published per year and their distribution over publication venues

- Searching for themes: we reviewed all the codes and sorted them into potential themes (e.g., management issues or socio-cultural issues). We used tables for visual representation of codes and themes, as suggested by Braun and Clarke [27].
- Reviewing themes, defining and naming themes: We performed the fifth and sixth steps in parallel. It included reading through all codes and revising their allocation to themes as well as reviewing all the themes and performing required modifications.
- Producing report: Our analysis revealed knowledge sharing challenges and practices of GSD teams are discussed around 6 main themes including: management, team structure, work processes/ practices, team cognition, social attributes and technology, which are elaborated in sections 6 and 7.

3. Demographic Attributes

3.1. Chronological View

Figure 2 represents the number of the selected studies published per year within the review period from 2000 to 2014. It is worth noting that we could not find relevant papers published in 2000, 2001 and 2003 and the review only covers the studies published before 7th September 2014, which is not a full year. Figure 2 reveals that the number of studies on knowledge sharing in GSD published per year has been increasing since 2007. One possible reason for this could be that the ICGSE (International Global Software Engineering Conference) conference started in 2006 and this conference and its co-located workshops raised the attention on knowledge sharing for distributed software engineering. We noticed that 39 studies (63.9%) were published during the last 6 years which means the role and application of knowledge sharing in GSD are receiving increasing interest and attention from researchers and practitioners.

3.2. Publication Venues and Types

An attempt to identify the types and venues for publications of a particular topic/theme can potentially be useful for researchers who may be interested in conducting research on a relevant topic. Figure 2 shows the types of papers (i.e., journals, magazines, conferences, workshops, and book chapters) included in this SLR. It is clear that Journal is the most popular publication type with 29 papers (i.e., 47.5%). Other papers have been published in conferences (20 studies, 32.7%), workshops (5 studies, 8.1%) and book chapters (4 studies, 6.5%), while only 3 studies came from magazines. The included 61 papers were published in 47 venues, in which ICGSE conference and Expert Systems journal are the leading venues for publishing work on knowledge sharing challenges and practices in GSD as they have published 11.4% (7 papers) and 4.9% (3 papers) of the included papers respectively. There are 6 venues with only two selected papers and it should be noted that 39 papers (i.e., 63.9%) were published in 39 different venues. The reviewed papers have been mainly published in three different research communities including Software Engineering (23 papers), Information Systems (21 papers) and Knowledge Engineering (5 papers). This finding demonstrates that researchers with different research interests have broadly considered this research topic.

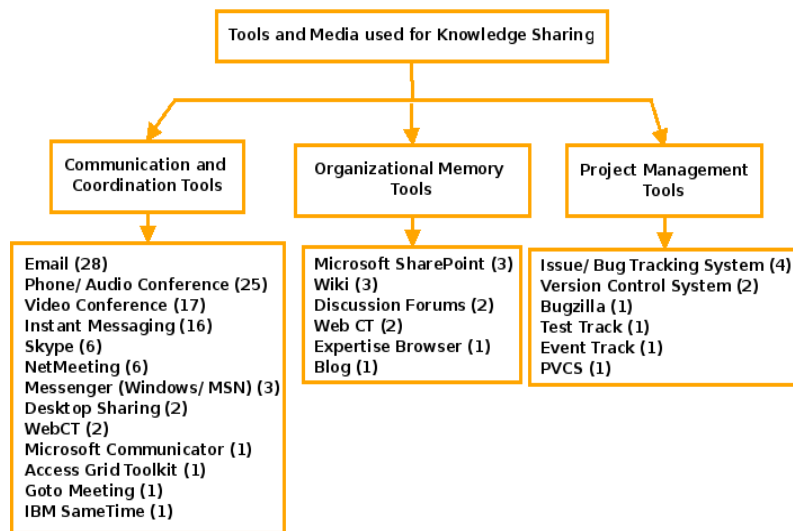


Figure 3 - Tools and Media for Knowledge Sharing

3.3. Tools and Media for Knowledge Sharing

The success of GSD projects heavily relies on communication media [28]. Hence, the choice of communication media can also help mitigate some of the knowledge sharing challenges. Therefore, we investigated the tools and media that had been reported in the reviewed papers. We counted the number of tools and media in each primary study because some primary studies conducted more than one case study. It is also worth noting that some study such as P51 in spite of conducting multiple case studies, stated a number of tools and media, which were used by all these case studies. For such kinds of study, we counted the proposed medium only once (i.e., counted based on primary study). We classified the tools and media into three groups: (i) communication and coordination tools; (ii) organizational memory tools; (iii) project management tools [29]. The first category, communication and coordination tools, focuses on improving awareness among GSD team members and enabling communicating using text, audio and video. It should be noted that a large number of studies only mentioned that teams/organizations use videoconferencing, audio-conferencing and instant messaging. In fact, they did not mention the name of communication tools. Organizational memory tools capture and document organizational knowledge e.g., what expertise people have [29], [30]. It is clear from Figure 3, despite the emergence of new communication media; email (28 papers) was the dominant communication medium in all selected studies. It is followed by telephone/audio conferencing (25 studies), video-conference (17 studies), Instant Messaging (16 studies), Skype (6 studies) and NetMeeting (6 studies). As shown in Figure 3, organizational memory and project management tools are less used for transferring knowledge by distributed team. These categories are not mutually exclusive, as there may be tools and media, which can be classified into more than one category. We only focused on what was reported as the main purpose of tool in each of the studies when categorising the identified tools. It should also be noted that most of the software development in general and knowledge sharing tools in particular have some features to support social interactions among distributed team members. When GSD team members use different tools and media applications, they get involved in social interactions with other team members. For example, most of the tools enlisted in Figure 3 can also be used for social interactions that can help built social ties, which are usually leveraged for knowledge sharing in GSD.

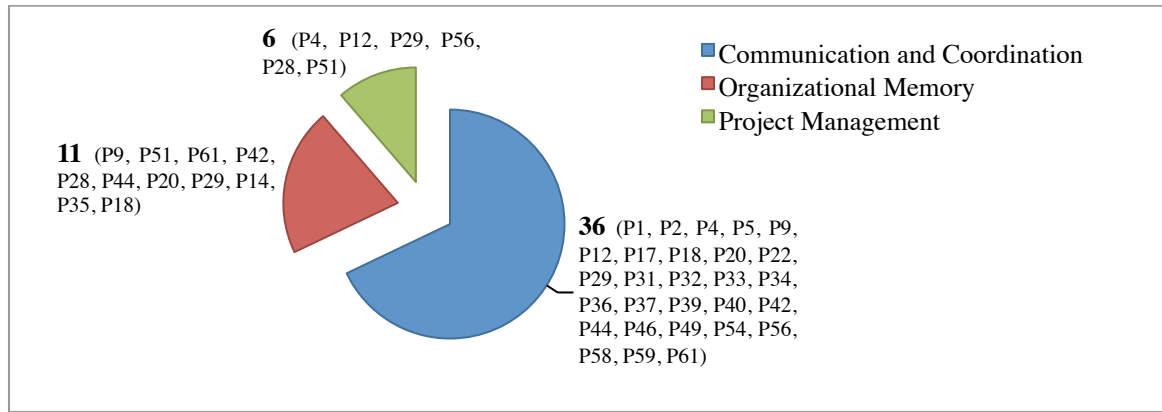


Figure 4 - Distribution of Primary Studies based on Types of Tools

4. Research Design Attributes

4.1. Research Methods

In order to identify the research methods reported in the selected papers, we did not employ a pre-defined classification of the research methods proposed in the literature [31]. We decided to classify the papers for the used research methods based on what was stated in the paper; for example, if authors claimed that they had used a case study research method, we classified that paper under the case study category. Table 4 shows the detailed information about the types of research methods used in the reviewed papers. It is clear that the case study is the most popular research method used in the reviewed papers as 43 papers (70.4%) used a case study research method. We also found that 5 papers (P12, P18, P20, P38 and P54) were based on ethnographic studies, which expect researchers to immerse themselves in the studied context and groups for an extended period of time applying a participant observer data collection approach [32]. Three studies (P10, P21 and P22) used controlled experiments. For instance, (P22) conducted a controlled field experiment in IBM to investigate processes related to technology use and internal knowledge sharing in co-located and distributed teams. There were 2 field studies (3.2%), 2 action researches (3.2%), 2 grounded theory (3.2%), 1 focus group discussion (1.6%), 1 survey (1.6%) and 1 design science (1.6%). We found that two papers (P11 and P24) did not indicate any research method. It is worth noting that one paper (P47) used more than one research method, in which a blend of design science and action research methodologies was used. That is why the sum in Table 4 (62) exceeds the number of the reviewed papers (i.e., 61). Of the 43 case studies, 7 studies used a specific type of case study methodology as their research methods, while the others did not state any specific type of case study. Two studies (P29 and P31) conducted explorative case studies. 5 studies (P5, P30, P32, P35 and P50) were longitudinal case studies. A longitudinal case study investigates the same case at several points in time in order to investigate how the case changes over time [32]. In (P4), it is shown that the issues around the nature of embedded knowledge, its management and consequences changed over time.

4.2. Data Analysis Types

The reviewed studies were classified into industry and academic cases. The academic category refers to those studies, which were performed in an academic setting; the industrial studies were carried out with industry. Table 4 shows that a majority of the studies (54 out of 61, 88.5%) are industry cases, whilst only 7 papers report academic cases. The results presented in this review have a significant level of relevance and practicality. Table 4 shows that a large majority (43 out of 61, 70.4%) of the papers used qualitative research approaches, while only 8 studies employed quantitative research approaches, in which all of them except two papers (P13 and P16) were situated in the academic category. It should be noted that 7 studies (i.e., P22, P45, P48, P53, P54, P56, and P60) used both qualitative and quantitative research approaches for data analysis and three studies (P23, P27 and P31) did not provide any information about data analysis.

4.3. Data Collection Methods

Table 4 shows that the most frequently used data collection approach was Interview (49 studies), followed by documentations/archival records (29 studies). We classified all the materials, documents, and artefacts

related to project and project members as documentations/archival records. In (P5), documents such as functional specifications, project plans, project reports, test cases and other documents related to software projects and work procedures were reviewed. Observation (21 studies), questionnaire (7 studies), workshop/discussion (6 studies) and focus groups (1 study) maintained the subsequent positions. There were only two studies (P23 and P27) in which they did not provide any information about the sources of data. Note that some studies may use more than one data source. That is why the sum in Table 4 (115) exceeds the number of the selected studies (61).

pre-Print Copy

Table 4 – An overview of research design attributes of selected studies

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24	P25	P26	P27	P28	P29	P30	P31	#	
Research Method (section 4.1)																																	
Case Study	♦		♦	♦	♦	♦		♦	♦				♦	♦	♦	♦	♦		♦				♦		♦	♦	♦	♦	♦	♦	♦	43	
Ethnographic Study												♦						♦		♦												5	
Controlled Experiment										♦											♦	♦										3	
Field Study																																2	
Action Research																																2	
Survey																																1	
Focus Group Discussion		♦																														1	
Design Sciences																																1	
Grounded Theory								♦																								2	
Unclear											♦													♦								2	
Data Analysis Types (section 4.2)																																	
Qualitative	♦	♦	♦	♦	♦	♦	♦	♦	♦		♦	♦		♦		♦	♦	♦	♦	♦				♦	♦	♦			♦	♦		43	
Quantitative										♦			♦	♦		♦					♦							♦				8	
Mixed																						♦										7	
Unclear																							♦				♦				♦	3	
Study Setting (section 4.2)																																	
Academic										♦				♦							♦								♦			7	
Industry	♦	♦	♦	♦	♦	♦	♦	♦	♦	♦	♦	♦	♦		♦	♦	♦	♦	♦	♦	♦	♦	♦	♦	♦	♦	♦	♦		♦	♦	♦	54
Data Collection Method (section 4.3)																																	
Interview	♦		♦	♦	♦	♦	♦	♦	♦		♦	♦	♦		♦	♦	♦		♦	♦		♦		♦	♦	♦			♦	♦	♦	49	
Archival Record			♦		♦					♦	♦	♦				♦	♦	♦				♦				♦			♦		♦	29	
Observation				♦	♦							♦						♦				♦				♦		♦	♦	♦		21	
Questionnaire														♦		♦					♦											7	
Workshop		♦											♦					♦														6	
Focus Groups																		♦														1	
Unclear																							♦				♦					2	
	P32	P33	P34	P35	P36	P37	P38	P39	P40	P41	P42	P43	P44	P45	P46	P47	P48	P49	P50	P51	P52	P53	P54	P55	P56	P57	P58	P59	P60	P61	#		
Research Method (section 4.1)																																	
Case Study	♦	♦	♦	♦				♦		♦	♦	♦	♦	♦	♦		♦	♦	♦	♦		♦		♦	♦	♦	♦	♦		♦		43	
Ethnographic Study							♦																♦									5	

4.4. Units of Analysis

Data collections described in Section 4.3 should be performed with respect to a well-defined unit of analysis [33]. A unit of analysis might be an organization, a project, a team, or an individual developer [33]. For some primary studies (e.g., those conducted multiple case study), the unit of analysis may be counted N times. For example, P29 studied two different projects through two exploratory case studies, and therefore we categorized its unit of analysis as project and counted it two times (i.e., two cases). We used the term “case” instead of “study” in this subsection. Among all the cases reported in the included papers, 63 cases have been selected project as the unit of analysis. Another unit of analysis commonly used in the reviewed papers was organization (37 cases). For example, (P53) conducted two case studies in two business units of Eco Company. These two units were referred to as BU1 and BU2 in which the development of the software used in both business units was distributed between the main site (i.e., Netherlands) and two remote sites: site A and site B. As shown in Figure 5, 11 papers used (distributed) teams as their unit of analysis. To give an example, Zahedi and Babar in (P43) studied a software development team distributed between Denmark and Pakistan forming an Extended Team Model (ETM) in order to understand what strategies are adopted to enable process knowing in GSD. We classified the unit of analysis of (P2, P7, P8, and P40) as individuals because these studies are based on experts’ interviews from different software companies. The summary of units of analysis used by primary studies is shown in Figure 5.

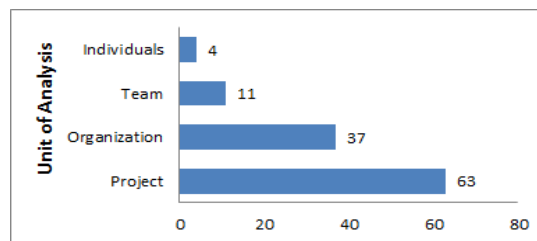


Figure 5 – Distribution of Units of Analysis used in Primary Studies

5. Contextual Settings

5.1. Collaboration Models

We classified the identified GSD collaboration models using the taxonomy proposed in [2]. This taxonomy has two key concepts: location and legal entity. Location can be divided into onshore and offshore. Onshore means sourcing is conducted within the same country, while offshore refers to sourcing in other countries. Legal entity shows whether the development is conducted within the same company or with another company. Insourcing means the development is conducted at another site of the company and outsourcing is with a different legal entity (i.e., leveraging external third-party resources) [2]. The combination of these concepts creates four collaboration models: 1) onshore-insourcing; 2) onshore-outsourcing; 3) offshore-insourcing; 4) offshore-outsourcing.

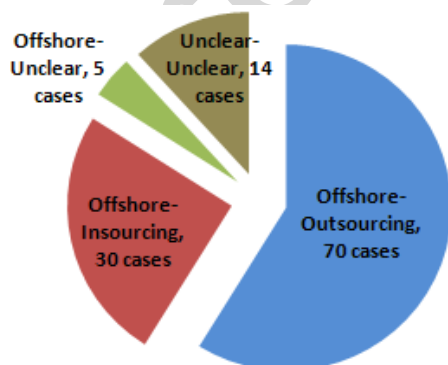


Figure 6 – Distribution of Studies over Collaboration Models

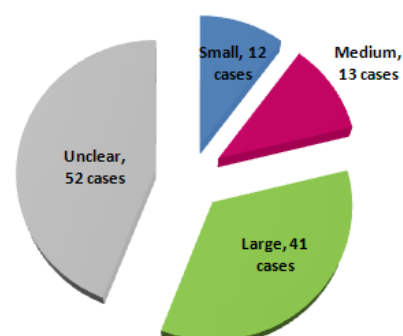


Figure 7 – Distribution of Organizational Size in Selected Studies

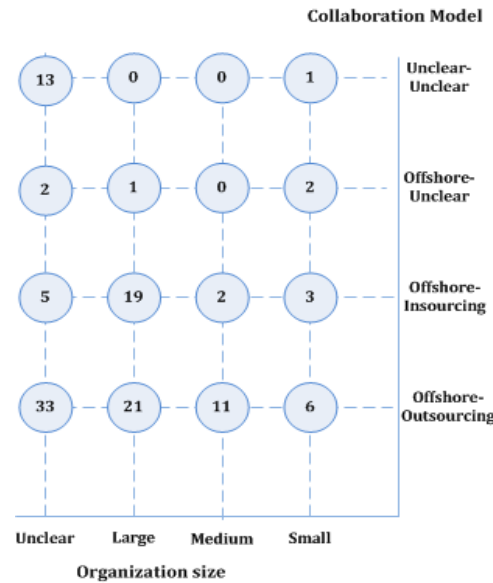


Figure 8 – Distribution of Studies over Collaboration Model and Organizational Size

Figure 6 shows the collaboration models used by primary studies. Since we did not have any paper with an onshore setting (i.e., teams are distributed within the same country), there is no onshore-insourcing and onshore-outsourcing collaboration model in Figure 6. According to Figure 6, the most common collaboration model used in the reviewed papers was offshore outsourcing (70 cases), followed by offshore-insourcing (30 cases). For example, we interpreted the collaboration model in (P32) as offshore-insourcing because there was an offshore development relationship between a British firm “Sierra” (a pseudonym) and its Indian subsidiary (i.e., headquarters in London and a subsidiary in Bangalore). We founded in 5 cases in 4 papers (P4, P12, P54 and P44) that it is not possible for us to recognize the legal entities between sites involved in GSD. Thus, we classified their collaboration models as offshore-unclear. It should be noted that we could not find any clear information about the locations of sites and legal entities between sites in 14 cases in 13 studies. It is worth noting that of these 13 studies, 7 studies (P10, P14, P21, P28, P35, P41 and P50) used students as subjects and two papers (P7 and P8) were based on experts’ interviews from different software companies. Therefore their collaboration models were not clear and we classified them as unclear-unclear.

5.2. Size of Organization

An organization’s size can play an important role in the types of knowledge sharing challenges faced and the potential strategies to address the challenges. For this reason, we decided to consider the size of the organization where the reported research was carried out. Figure 7 and 8 show the results. We classified the organization size into small (<100 employees), medium (<1000 employees) and large (>1000 employees). For the offshore-outsourcing collaboration, we only considered the size of vendor organizations. We found that 52 cases (44%) failed to report any information about organizational size. For those cases, which reported the organization size, large organizations dominate (41, 34.7%) over medium (13, 11%) or small organizations (12, 10.1%). For example, (P49) conducted a qualitative case study at a large German multinational company, which had more than 300,000 employees worldwide and over 18000 employees in India. It specifically examined the impact of social capital on effectiveness of knowledge transfer between German onshore headquarters and its Indian subsidiary. Therefore, we classified this case as an offshore-insourcing model in a large organization.

Some cases did not provide any information about the number of employees in the organization and they only claimed that the size of organization is large, medium and small. In these cases, we categorized the organization size based on authors’ claims. In particular, for well-known organizations, the studies (e.g., P22 conducted in IBM) only provided the name of the organization. In this situation, we have tried to examine other available resources (e.g., organization’s website) for getting the information about the size of well-known organizations. In this way, we could make our interpretations and inferences as reliable as possible.

5.3. Locations and Number of the Sites involved in GSD

The reviewed studies mentioned 41 different countries from where companies were involved in GSD. Table 5 shows that the most frequently mentioned countries are India (57 cases), Unites States (42), Germany (24 cases), Netherland (11 cases) and Russia (8 cases). The collaborations between USA and India were reported in most of the reviewed papers. The Asian countries such as India, China and Vietnam rarely acted as clients as they are very popular destinations for outsourcing. Our findings reveal that the Eastern European countries such as Ukraine, Serbia and Belarus are emerging as new destinations for outsourcing. One possible reason for this could be the existence of a low-cost workforce in such countries.

We also collected the information about the number of involved distributed sites in each case in the reviewed papers. Table 6 reveals that the majority of the distributed collaborations were conducted between 2 sites (82 cases), followed by 3 sites (20 cases). There was one case with 4 sites, two cases with 5 sites, one case with 6 sites and two cases with 13 and 15 sites respectively. In (P9), the authors investigated knowledge sharing barriers and enablers through a case study of 2 firms (firm A and B). There was offshore-insourcing collaboration between 13 sites (i.e., countries) of firm A, while the firm B established its offshore-insourcing collaboration model within 5 sites. We found that 11 cases did not say anything about the number of sites.

Table 5 – Locations involved in GSD

Name of Country	#	Name of Country	#	Name of Country	#
India	57	Pakistan	4	Italy	1
USA	42	Ukraine	3	Spain	1
Germany	24	Vietnam	3	Costa Rica	1
Netherland	11	Brazil	3	Taiwan	1
Russia	8	Norway	3	Israel	1
China	7	Serbia	3	Romania	1
Ireland	6	Australia	3	Finland	1
New Zealand	6	Belarus	2	Philippines	1
Switzerland	5	Argentina	2	Hong Kong	1
Denmark	5	Mexico	2	Singapore	1
UK	5	Malaysia	2	Chile	1
Canada	5	Japan	2	Oman	1
Sweden	4	Iran	1	Peru	1
Poland	4	Colombia	1		

Table 6 - Studies distribution over number of sites and collaboration models

Collaboration Model	Number of Sites						Unclear
	2 sites	3 sites	4 sites	5 sites	6 sites	>6 sites	
Offshore-Outsourcing	55	9	1	1	0	0	4
Offshore-Insourcing	17	7	0	1	1	2	2
Offshore-Unclear	3	2	0	0	0	0	0
Unclear-Unclear	7	2	0	0	0	0	5

6. Knowledge Sharing Challenges in Global Software Development

This section reports the results from analyzing the data about the knowledge sharing challenge. This analysis was meant to answer RQ1, “*What are the key challenges of sharing knowledge in global software development?*” As we discussed in subsection 2.5.2, we employed the thematic analysis method to analyze this data item. Table 7 shows the knowledge sharing challenges in GSD reported in the reviewed papers.

6.1. Management

Under this category we present the challenges that are associated with managerial commitment and actions towards knowledge sharing in GSD.

Cost of Knowledge Sharing. Knowledge sharing in distributed settings is associated with high cost (P2, P8, P9, P12, P48, P42, P61) that might not be known beforehand (P8, P48). The knowledge sharing cost can fall under different categories such as the budget for travelling between sites (P2, P8, P9, P48), cost of virtual communication (P8, P61) and extra effort for codifying knowledge (P12, P2, P42). Dingsøyr and Smite (P2) found the need for extra budget was an inhibitor to form virtual communities of practice between sites. Nevertheless, in the absence of virtual communities much of the project knowledge remains tacit at each site. Manteli et al. (P61) found that software developers from the vendor’s side did not have full access to knowledge sources (e.g., documentation) due to the client’s business policies. They (P61) observed increased time and effort were required for creating, storing, and sharing filtered knowledge with the vendor’s team.

Employee Turnover. We found several studies (P19, P32, P44, P48, P49, P51) indicating that employee turnover in distributed settings highly disturbs the knowledge-sharing process. Staff attrition enforces extra energy for building social relationships between sites (P32, P49). It introduces gaps in the knowledge sharing process (P19, P44, P51, P49) and causes sharing project knowledge repetitively with newcomers (P48, P49, P32), which is not only time-consuming but also de-motivating for knowledge sources (P49). The authors of (P49) found that a high rate of turnover at offshore was a major inhibitor to knowledge sharing activities. While the company was investing on bringing Indian employees to onshore (Germany) for training, building social relationships and developing contextual awareness, losing trained members was causing network instability and enforcing repetition of process. Besides, a high rate of staff turnover at offshore impacted willingness of onshore members to spend time with the remote team for sharing knowledge.

Low Priority Perception to Knowledge Sharing Activities. Management giving knowledge sharing low priority is considered one of the key challenges. It was evident that knowledge-sharing activities were ranked as low-priority (P30, P46, P47) that could be sacrificed (e.g., cutting allocated time and budget) when the team faced harsh deadlines. This approach could also influence attitude and responsiveness of team members towards remote colleagues. Given the lack of structured plan for knowledge transfer, it is observed that onshore members considered answering questions/ emails of offshore team as a second-priority and focused on own daily tasks in accordance with deadlines (P30).

6.2. Team Structure

Vague Definition of Roles and Responsibilities. It has been reported that unstable organizational design and vague definition of roles and responsibilities (P6, P28, P39, P61) create flaws in proper sharing and seeking knowledge in distributed teams. It causes confusion in identifying who is supposed to decide or take action on certain tasks (P28, P6). We have found that lack of clear division of work between sites blurs task ownerships and reduces motivation of dispersed members to proactively seek and share knowledge (P39).

Hierarchical Structures. Our review has revealed that hierarchical leadership causes bottlenecks in knowledge sharing (P61) and impacts the flow of information (P36, P12). Boden et al. (P12) reported that the client team applied top-down control mechanisms to ensure the quality of requirement specifications produced by the vendor team. Nevertheless, clients did not recognize that the root cause of poor specifications was lacking domain knowledge at offshore. This type of controlling approach discouraged knowledge sharing needs, increased frustrations at both ends, and contributed to the termination of distributed collaboration.

Table 7 - Knowledge Sharing Challenges in GSD

	Challenge	Key Points	#
Management	Ch1: Cost of Knowledge Sharing	<ul style="list-style-type: none"> Budget for travels between sites (P9, P48, P8, P2) Cost of Virtual communication (P8, P61) Cost of Codifying knowledge (P12, P2, P42) Extra effort for managing filtered information sharing (P61) 	7
	Ch2: Employee Turnover	<ul style="list-style-type: none"> Gaps in sharing and understanding knowledge (P19, P44, P51, P49) Need of re-building social relations (P32, P49) Need of re-transferring knowledge to new comers (P32, P48, P49) Lack of motivation to share knowledge with newcomers (P49) 	6
	Ch3: Low Priority Perception	<ul style="list-style-type: none"> Sacrificing knowledge sharing activities to meet project deadlines (P30, P46, P47) 	3
Team Structure	Ch4: Vague Role definitions between sites	<ul style="list-style-type: none"> Difficulty to locate source of knowledge due to different role descriptions (P61) Sub-optimal flow of information due to unclear roles and responsibilities (P28, P39, P6) Reduced motivation to share knowledge due to vague ownerships (P39) 	4
	Ch5: Hierarchical Structures	<ul style="list-style-type: none"> Bottlenecks in knowledge sharing due to hierarchies (P61) Negative impact of higher social standing of brokers on knowledge sharing and learning behaviours (P36) Handicapped articulation and clarifications between sites due to top-down leadership (P12) 	3
Work Processes/ Practices	Ch6: Documentation Problems (missing, poor, outdated)	<ul style="list-style-type: none"> Poor documentation of requirements for remote team (P26, P45) Difficulty to find accurate source of information due to missing/ out-dated documents (e.g. requirement, design decisions) (P28, P61) Poor organizational memory due to lack of relevant documentation (P46, P47) 	6
	Ch7: Shortcomings in maintaining group awareness	<ul style="list-style-type: none"> Difficulty to understand task status between sites (P26) Difficulty to understand presence status and activities of distributed collaborators (P11) Difficulty to understand task interdependencies between sites (P44) Gaps in understanding who knows what (P2) Gaps in understanding changes during development work (P56) 	5
	Ch8: Communication Challenges due to Distance	<ul style="list-style-type: none"> Difficulty to create informal networks due to lack of face-to-face contact (P8) Reduced communication frequency due to distance (physical, temporal) (P9, P8, P6, P16) Overload to share knowledge from physical distance via media (P30) 	5
Team Cognition	Ch9: Contextual Difference	<ul style="list-style-type: none"> Difficulty to identify knowledge to transfer (P30, P8, P46, P5) Difficulty to articulate tacit knowledge (P5, P6, P49) Difficulty to mobilize and apply knowledge in other context (P32) 	7
	Ch10: Gap in Education and Technical Knowledge	<ul style="list-style-type: none"> Reduced quality of knowledge transfer due to technical knowledge imbalance (P5, P17) Difficulties to communicate due to lacking common technical terminologies (P5) Limited understanding of inexperienced workforce (P13, P42, P37) 	5
Social Attributes	Ch11: Lack of Trust and Rapport	<ul style="list-style-type: none"> Lack of social ties to informally share knowledge (P9, P45, P32, P38) Lack of trust in competencies of remote members (P38, P42) Low commitment trust to knowledge sharing activities (P38) 	5
	Ch12: Fear	<ul style="list-style-type: none"> Unwillingness to share knowledge due to fear of losing job (P13, P19, P37, P49) Fear of losing competencies due to lacking knowledge backflow to client (P8) 	5
	Ch13: Lack of Openness	<ul style="list-style-type: none"> Reluctance to ask questions (P13, P22, P42, P48, P8, P37, P19) 	7
	Ch14: Linguistic Distance	<ul style="list-style-type: none"> Difficulties to share and absorb knowledge due to language difference (P9, P19, P8, P58) 	4
Tools	Ch15: Limitations of Tools for Knowledge Sharing	<ul style="list-style-type: none"> Need of Collaborative Tools (P4) Difficulties to use Search Functions in Knowledge Repositories (P2) 	2
	Ch16: Shortcomings in Utilizing existing Tools	<ul style="list-style-type: none"> Lack of Central Knowledge Repository (P40) Not using/ populating Knowledge Repositories by Team (P2, P24) 	3

6.3. Work Process

Documentation Problems. Several studies (P26, P28, P45, P46, P47, P61) highlight the impact of the lack of proper documentation on sharing knowledge in GSD. The use of Agile approaches and emphasis on tacit communication could negatively impact producing and maintaining explicit knowledge (P61, P28). One study (P28) reported that due to agility much of project knowledge (e.g., requirements, design decisions) remained fluid and scattered in source code or test cases. Staple and Schneider (P28) argued that documents were inconsistent with fluid information, which caused misunderstanding for distributed development teams locating accurate source of knowledge. We found that abstract requirement specifications were not sufficient for offshore teams and could cause misunderstanding (P26, P45). Rottman and Lacity (P45) observed the client process was highly reliant on social interactions, in which gathering and analyzing requirements was rather informal by walking to business users and clarifying questions (i.e., referred as “sneaker-net”). They (P45) observed requirements were informally documented on whiteboards and personal notebooks; this approach was found to be unsuitable and prohibited proper knowledge transfer, as it was impossible for offshore teams to make social relations with business users in US to clarify requirements (P45).

Shortcomings in Maintaining Group Awareness. Lack of awareness indicates breakdowns in sharing relevant knowledge between distributed team members. Espinosa et al [34] define “awareness” as short-term knowledge that can exist about teams (e.g., who is involved in a project) and the tasks (e.g., status of tasks). Lack of awareness may stem from different reasons such as different organizational cultures (P56) and shortcomings in communication and coordination mechanisms between sites (P44, P56, P26, P2). It can cause gaps in the understanding of distributed team members from aspects such as changes during development (P56), dependencies of tasks (P44) and progress status (P11). Some studies (P26, P2, P44) underlined the role of status meetings in maintaining group awareness. It has been reported that organizing status meetings locally (P2) hampers sharing knowledge about team members and introduces gaps in understanding of “who knows what” in the team. Authors of (P44) reported that frequent absence of dispersed members from joint meetings brought a gap in understanding the status and interdependencies of tasks undertaken by individuals.

Palacio et al. (P11) discussed that synchronous collaboration of GSD teams requires provision of certain information about members to the team. They argued that lack of awareness about team members such as who is assigned to the same project, members’ presence status and activities they are performing introduce challenges on the flow of information and synchronous collaboration of dispersed teams.

Communication Challenges due to Distance. Some studies considered knowledge sharing is challenging in GSD due to well-known fact of “distance” (P6, P8, P9, P16, P30). Physical distance prohibits “face-to-face” interactions (P8), poses critical issues to synchronous communication (P6, P8) and hardens the formation of informal networks (P8). Due to temporal distance GSD teams experience a limited time window to synchronously communicate and collaborate with remote colleagues (P9, P8). Analysing requirement-centric social networks between GSD teams, it is found that distance does impact frequency and ease of communication between remote colleagues to share knowledge about requirements (P16).

6.4. Team Cognition

We found several studies discussing that GSD introduces knowledge gaps between sites that are challenging to deal with. When the knowledge gap is significant it could impact effectiveness of knowledge sharing in terms of comprehensiveness and understand-ability. We elaborate on the challenges classified under this category.

Contextual Difference. In addition to other distance dimensions (e.g., geographical, temporal, linguistic), distributed teams experience contextual distance that influences their capability in sharing knowledge and making common understanding [35] [36]. Sole and Edmondson [36] draw attentions to knowledge that is embedded in the work practices of organizations and shared by those who are collocated- called *situated knowledge*. They argued that situated knowledge is associated to location and varies in different social and physical contexts; hence, it is difficult to be codified and shared with individuals who are in different contexts. Similarly, Hinds and Weisband [35] discuss that people working in different contexts are likely to neglect sharing relevant information that could help remote co-workers to understand the environment in which they are working. It is because they may not be able to identify what information is missing in other locations that needs to be mentioned.

Our review revealed several studies that report the difficulties to share and understand contextual information (P6, P32, P49) and specifically business domain knowledge (P5, P8, P30, P46) in GSD teams. It has been found that mobilizing knowledge from headquarters to subsidiaries was problematic due to embedded-ness in context (P32). Business domain knowledge (e.g., banking regulation) could vary from country to country (P8, P5, P49), which is difficult to be identified in detail (P8, P5, P30, P46) and articulated in specifications (P5, P6, P49). However, lack of understanding from domain knowledge impacts the capability of remote teams to make proper interpretation of requirements that is needed for proposing technical solutions (P8, P46, P49, P5). Zimmermann and Ravishankar in (P49) found that unshared contextual understanding of dispersed teams from requirements, software and application domain was challenging. They (P49) argued that the Indian team needed a high level of domain knowledge (i.e., automotive engine) to understand how software applications fit into the final product; yet comprehensively sharing such knowledge was difficult for onshore team and interpreted as insufficient specifications offshore.

Different Education and Level of Technical Skills. Knowledge gaps could be caused by technical knowledge imbalance between dispersed teams due to different experiences and educational backgrounds (P5, P13, P17, P37, P42). Technical knowledge imbalance could hamper quality of knowledge sharing (P5, P17) due to the limited absorptive capacity of knowledge recipients (P13, P42, P37) and lacking common ground to encode and decode exchanged messages (P5). One study (P37) reported that the steady increase in the size of offshore teams introduced a significant technical knowledge gap between the two sites because a majority of the new comers were graduates, who did not have the experiences required for undertaking complex software testing within virtual teams. Casey and Richardson (P37) described that a large technical gap created difficulties for onshore team to have a realistic understanding of the expertise at remote site. The situation was aggravated by cultural differences (e.g., willingness to say yes by the offshore team).

6.5. Social Issues

Social interactions play an important role in knowledge sharing. Several of the reviewed studies (P9, P32, P38, P42, P45) reported lack of trust and rapport prohibited the emergence of social networks and informal communication channels that are necessary for supporting knowledge sharing in GSD. One study reported that there could be three types of lack of trust (P38): lack of trust in competency of remote team members, lack of companion trust (i.e., difficulty to identify a member with relevant knowledge, or if known, difficulty to establish contact) and lack of commitment trust (i.e., being truly committed to knowledge sharing).

Several studies have found that dispersed members were unwilling to share knowledge with remote colleagues due to fear of losing jobs (P13, P19, P37, P49) and competencies (P8). It was clear that management's emphasis on globalization could have a negative impact on the motivation and cooperation of local team members to share their skills with new comers, as they felt insecure about their future.

GSD poses difficulties for expressing messages and absorbing knowledge between dispersed teams because of linguistic differences (P8, P9, P19, P58). One of significant impacts of social issues on knowledge sharing can be lack of openness and clarity of interactions between sites (P8, P13, P19, P22, P37, P42, P48). That can be interpreted as feeling embarrassed to seek inputs or share ideas that may reveal lack of skills (P22). Lack of openness inhibits the provision of feedbacks and clarification of expectations between dispersed teams (P13) and can lead to misunderstandings from requirement specifications (P8).

6.6. Technological Issues

Some of the reviewed studies also reported knowledge sharing challenges caused by technological problems. Two studies described the technological limitations inhibitors to knowledge sharing in GSD (P4, P40). In (P4), authors highlighted the difficulties in sharing spatial knowledge (e.g., tacit knowledge generated around Kanban board) in distributed settings due to a lack of suitable tools for visualization and synchronous collaboration. Also, difficulty to effectively use search functions for retrieving information from knowledge repositories for sharing knowledge was reported in (P2). Other studies in this category, however, noted lack of strategies/plans for effectively applying existing tools. In (P24), authors reported that Blogs and Wiki pages either remained unpopulated or did not have the required information. The findings from (P40) showed that distributed members were facing difficulties to locate the source of architectural knowledge as there was no central repository; key architectural artefacts were scattered on Wiki pages, SharePoint folders, network drives and emails which were not always updated. One study indicated the tendency to store plans, progress reports, retrospectives and feedbacks locally, rather than sharing them through a central repository (P2).

7. Knowledge Sharing Practices in Global Software Development

This section reports the findings from analysis of the data extracted to answer RQ2, “*What are the practices and mitigation strategies that are proposed to support knowledge sharing in GSD?*” Table 8 provides an overview of the knowledge sharing practices reported in the reviewed papers.

7.1. Management

Incentives and Motivation. The reviewed studies reported that management’s strategies could be very important for promoting knowledge sharing. Mathrani and Parsons (P52) concluded that rewarding team members and changing bureaucratic process of the organization to adhocratic processes could improve the flow of knowledge between distant members. Gregory et al. (P19) observed that smoothly restructuring a team (i.e., no firing but replacing retired workforce with offshore resources) was helpful to reduce fear of losing jobs and stimulating motivation of client members to share knowledge. Madsen et al. (P30) suggested making effort and effect of knowledge sharing transparent by strategies such as identifying participants and estimating required time (e.g., for answering questions). Transparency and planning could motivate team members allocating time to knowledge transfer activities and deal with priorities (P30).

Temporary Collocation. Organizing visits to remote sites is one of the most prominent practices reported by several studies to support knowledge sharing across dispersed teams (P1, P7, P8, P9, P12, P17, P18, P19, P26, P28, P29, P40, P43, P49, P59). Temporary collocation of distributed team members gives rise to effectively sharing implicit knowledge. It helps the visitors to understand context and leverages mutual learning (P1, P7, P18, P59, P46). Collocation can be a crucial means for learning “how-to” knowledge specific to a location. Furthermore, visiting remote sites and experiencing face-to-face interactions significantly help distant members to bridge cultural gaps and develop social relationships, which accordingly stimulate knowledge sharing (P9, P12, P19, P26, P40, P49, P59).

Organizing remote site visits vary in terms of duration, frequency and the stage that it takes place in offshoring collaboration (P1). Sometimes the visits may last for longer periods of time, so-called “*rotating members between sites*” (P7, P28, P26, P17). Rotating roles between sites found to be helpful to promote business domain knowledge in remote location, as onsite members usually possess higher domain knowledge due to direct interaction with customers (P7). Stapel and Schneider (P28) suggest sending contextual ambassadors to remote sites to act as context-sharing catalysts. Some companies in GSD engagement implement more specialized remote visits in the form of collocated project kick-off (P7, P8, P29, P39, P40). In the context of distributed agile development, Dorairaj, Noble and Malik (P7) referred to organizing “inception workshops” before starting a project with participation of stakeholders and key development team members; these interactive and structured workshops were crucial to build shared vision on project goals and management aspects such as schedule, technology assumption and rules of engagement.

7.2. Team Structure

Flexible Communication Structure. Flexibility in communication structure in GSD facilitates knowledge sharing. Removing hierarchical hurdles (P32) and enabling flat communication pattern (P12) can ease the flow of information at different levels in GSD teams. It assists one-to-one communication between dispersed members and provides more opportunities to share tacit knowledge through informal chats. Boden et al. (P26) underlined the benefit of having direct access to customers from offshore in order to deal with difficulties of sharing and understanding requirement specifications. Direct communication link to customers can enable offshore to get engaged in discussion, elicit requirements from customers and prepare specifications themselves.

Clarifying Work Structure. A set of studies (P9, P10, P19, P40, P43) reported that a good clarification of the work structure could help smooth the flow of information between distributed team members. In (P10) the authors present results of a controlled experiment ran with two groups of distributed student teams. While roles, responsibilities and assigned tasks were clearly defined for one of the groups, the other group was asked to work on the same assignment only using joint team effort. The results of the experiment showed that distributed teams with clearly defined roles and responsibilities experienced better exchange of knowledge during project lifecycle and developed higher understanding of requirements (P10).

Formal definition of roles and assignment of responsibilities in (P19) were considered helpful to clarify ownerships, monitor performance on the individual basis and control successful knowledge exchange between sites. It was also reported that mirroring organizational design of the client at the vendor site was

significantly helpful to share knowledge about the business domain and processes; as the vendor team knew whom to ask when they need information on critical business issues or methodological dependencies (P19). Defining and/or improving the software development process (P9, P43) were also considered helpful to organize the sharing knowledge. As discussed in (P43), company managers found it crucial to develop common understanding of process between sites. They believed that it was required to avoid coordination breakdowns in project phases, increase visibilities and raise responsibilities of individuals to take ownership of tasks and proactively seek/ share information with remote colleagues (P43).

Table 8 - Knowledge Sharing Practices in GSD

	Practice	Key Points	#
Management	PR1: Temporary Collocation	<ul style="list-style-type: none"> Visiting remote sites to build social relation and enable team interaction (P49, P40, P26, P19, P9, P12, P59) Temporary Collocation to work on limited number of tasks (P43) Rotating Roles to distribute knowledge between sites (P7, P28, P26, P17) Temporary Collocation for Training and Situated Learning (P46, P59, P1, P18, P7, P49) Collocated Kick-off to create shared vision about project and members (P7, P8, P40, P39, P29) 	17
	PR2: Incentives and Motivations	<ul style="list-style-type: none"> Rewarding Members for sharing Knowledge (P52) Smooth Restructuring to mitigate Fear of losing jobs (P19) Making KT efforts transparent to motivate participation (P30) 	3
Team Structure	PR3: Flexible Communication Structure	<ul style="list-style-type: none"> Ease of information flow due to flat structure (P12, P32) Higher understanding of requirements due to access to customer from offshore (P26) 	3
	PR4: Clarifying Work Structure	<ul style="list-style-type: none"> Clear definition of Roles and Responsibilities (P10, P19, P40) Defining Work Process (P9, P43) 	5
	PR5: Using Boundary Spanning Roles	<ul style="list-style-type: none"> Bridging communication gap by mediating language / culture (P3, P19, P26, P20, P13, P57) Facilitating Communication by building social relation between sites (P3, P45) Bridging knowledge gaps (domain/ technical) between sites (P3, P29, P20, P36, P57, P8) Building common grounds in regard to coordination issues (e.g. deadlines, deliverables) (P20) 	10
	PR6: Forming Virtual Communities	<ul style="list-style-type: none"> Shaping Communication channel via cross-site communities (P39) Virtual Architectural Council to promote knowledge exchange and balancing decision making power between sites (P40) Virtual cross-functional communities to bridge communication gap between functional teams (P43) Communities to share knowledge on specialized discipline (P7, P44) 	5
Work Process/ practice	PR7: Frequent Communication	<ul style="list-style-type: none"> Regular joint meetings to maintain group awareness (P18, P19, P7, P4, P8, P22, P44, P26, P29, P40, P17) High communication volume between individuals (P14, P40) 	12
	PR8: Documentation	<ul style="list-style-type: none"> Explicitly defining business terms, rules and their relation to entity attributes (P27) Systematic review and revision of specification to assure quality of inputs (P29) Maintaining detailed artifacts (e.g. specification) to avoid misunderstanding between sites (P42) Codification to bridge knowledge boundaries between distributed stakeholders (P55) Codifying project knowledge (e.g. design) for organizational memory (P29, P7) 	5
	PR9: Joint Work between Sites	<ul style="list-style-type: none"> Engaging in Knowledge sharing via jointly defining and assigning tasks (P23) Engaging in knowledge sharing via Question-driven task scheduling between sites (P23) Engaging in Knowledge sharing via joint process roll-out (P43) Engaging in Knowledge sharing via jointly defining KPIs (P43) Integrating collaborative knowledge via Pair programming (P7) Facilitating learning in practice by Joint participation in project phases (P12) 	4
Team Cognition	PR10: Fostering Transactive Memory (TM)	<ul style="list-style-type: none"> Codifying knower (e.g. code comments) to link expertise and facilitating knowledge integration (P55) Maintaining codified knowledge directory to establish TM and enable knowledge exchange (P34) Providing skills database for visibility of expertise and knowledge of “who knows 	7

		<ul style="list-style-type: none"> what” between sites (P37, P18) Social interaction between members to establish TM and facilitate knowledge integration (P39, P54, P34) Governance strategies (e.g. tightly coupling tasks) to strengthen TM and ease of knowledge sharing (P53) 	
	PR11: Identifying gaps and Verifying Understanding	<ul style="list-style-type: none"> Reverse Presentation Technique (P25, P42, P8, P19, P46) Storytelling Technique (P6) Explication of conventions by providing Regular Feedbacks to offshore (P23) Diagnosing knowledge gaps between sites (P5, P30) Flexible short-cycle Iteration to understand business requirements (P38) Written/ Oral Tests after Knowledge Transfer sessions (P46) 	10
	PR12: Improving Team Qualification and Expertise	<ul style="list-style-type: none"> Mentoring offshore by onshore experts on domain and technical skills (P37, P46) Training to utilize Knowledge Repositories (P24) Training on communication within the team (soft skills) (P24, P7) Training on Technical Skills (e.g. programming, agile method) (P7) Providing simulated problems to offshore for practicing solutions (P46) Utilizing highly qualified team members (P9, P37) 	5
Social Attributes	PR13: Social Ties (Trust and Rapport)	<ul style="list-style-type: none"> Higher extent of knowledge exchange through trusted personal network (P14, P24, P50) Accelerated knowledge sharing through social ties (P9, P4) Higher understanding from shared knowledge due to social ties (P9, P49) Improved ability and willingness to share knowledge due to social ties (P49, P54) 	7
	PR14: Team Cohesion	<ul style="list-style-type: none"> Effectiveness of knowledge transfer due to shared identity (P35, P38, P49) Higher degree of knowledge transfer due to collectivist culture (P14) 	4
Tools and Techniques	PR15: Providing Groupware	<ul style="list-style-type: none"> Knowledge Repositories as reference point for sharing explicit knowledge (P44, P7, P2, P8, P59, P60, P42, P29, P22, P15) Communication media for sharing knowledge (synchronously and asynchronously) from distance (P8, P31, P39, P29) 	12
	PR16: Novel Techniques	<ul style="list-style-type: none"> Traceability Model Solution (P33) Experience Base Right Management Solution (P41) By-Product Technique for Documentation (P28) Flow Mapping Technique to Capture Information Flow (P28) Framework for choosing suitable Groupware tools (P21) Knowledge Flow Identification (KoFI) Methodology (P11) 	5

Using Boundary Spanning Roles. Several of the reviewed studies (P3, P8, P13, P19, P20, P26, P29, P36, P45, P57) reported the use of “boundary spanner” (i.e., also called knowledge broker, liaison role (P57), human bridge (P25) or system engineering bridge (P3)) to facilitate knowledge sharing in GSD. Boundary spanning in context of GSD referred to a “*capability of individual project members to broker and intermediate the relationship between client and vendor by assuring the effective information and knowledge sharing between team members*” (P19). This role could be officially assigned to project members (P3, P19, P29) or informally assumed by volunteers (P26). The brokers facilitate knowledge sharing by bridging communication gaps (P3, P13, P19, P26, P45) as well as knowledge gaps (P3, P19, P20, P26, P29). It is evident from the reviewed studies that individuals, who are experts in business domain and/or technical skills, usually take the intermediating role between the sites for helping to share domain and technical knowledge effectively between locations. In (P3), researchers underscored the role of system engineering bridges taken by external consultants to facilitate communication and collaboration between Japan and Vietnam. In that case, mediators actively participated in knowledge creation, requirement gathering and producing specifications. They significantly helped to bridge communication gaps and common understanding of requirements by sharing knowledge through documentation and social interactions with both ends by frequently travelling between locations and regularly being in touch with the dispersed team members. Locating vendor’s technical lead at the client’s site can help mitigate communication barriers for improving knowledge sharing (P13). Two more studies reported the benefits of locating vendors’ project managers to the clients’ site to facilitate extensive communication about critical business and functional issues between client and vendor teams (P19, P20). Furthermore, boundary spanners can also be helpful to improve social relationships by fostering mutual trust and confidence between locations (P3, P20) by enabling informal communication channels (P19) and facilitating sharing implicit knowledge (P20) between sites. One study (P45) concluded that one of the best mechanisms for building sustainable social capital is to

invest in onsite members as the relations built through this mechanism can have significant impact on intellectual capital (e.g., understanding requirements) and a project's throughput (P45).

Forming Virtual Communities of Practice. Another noticeable practice to facilitate knowledge sharing between dispersed team members is forming virtual communities (P7, P39, P40, P43, P44). Such virtual communities have been reported to be helpful to ensure adequate communication taking place between distant parties and raising awareness during the project lifecycle. In (P40), Clerc, Lago and van Vliet observed the formation of a virtual team including team members from all locations acting as the architectural council. That community helped in exchanging critical architectural information between sites and balance decision-making power. Similarly, in (P43) virtual cross-site communities (e.g. so-called "What-Team") were formed by participation of representatives of different functional teams and ensured all the related stakeholders were kept in the loop during different phases of the project.

7.3. Work Process Practices

Frequent Communication between Sites. Consolidating frequent interaction between distributed teams is a well-known practice to enable knowledge sharing (P14, P17, P29, P40, P7). A study (P14) reports that the more individuals have communication with remote members, the more knowledge they transfer to a distant site. Kristjansson et al. (P29) recommended maintaining frequent interactions in distributed teams on a daily basis around functional and technical issues. They emphasized that on-going communication and collaboration of sites was necessary to achieve full understanding of specifications.

Several studies (P4, P7, P8, P18, P19, P22, P26, P44) indicate that regular joint meetings help circulate knowledge within a distributed team and maintain group awareness on different aspects of projects such as task status, issues, schedules etc. Studies such as (P4, P7) argue for organizing regular scrum meetings with remote members to synchronize tasks in distributed settings. In (P4), it was found that daily scrum meetings were held via Skype where the team leader distributed tasks and shared knowledge through a brief discussion. These informal conversations were also automatically stored in the form of chat history. Researchers in (P40) observed high communication volume with distant sites facilitated through video-conferencing, instance messaging (IM), email and phone calls. The authors (P40) also underscored the crucial role of ad-hoc communication between sites (e.g. through emails, IM) in addition to scheduled frequent meetings to handle urgent situations and minimizing delays in the process of seeking and sharing the required information.

Joint Work between Sites. Our literature review identified the practices that emphasize close collaboration between sites fostering knowledge co-creation (P7, P12, P23, P43). In this regard, Boden et al. (P12) observed that joint participation of dispersed team members in a project's all phases (e.g., specification, development, and testing) facilitated learning by doing in the team. In addition, we identified several practices such as question-driven knowledge transfer (P23), joint task planning (P23), joint process rollout (P43) and pair programming (P7) that enable distributed team members to closely collaborate with each other and getting engaged in knowledge co-creation.

Documentation. Several of the reviewed studies (P7, P27, P29, P42, P55) reported the implementation of codification practices for explicitly sharing knowledge in GSD. Kotlarsky et al. (P55) argue that codification practices are helpful to coordinate expertise between different actors (i.e., client, onsite and offshore) in distributed settings and bridging knowledge boundaries. It has been reported that knowledge codification helps reduce interdependencies between the client's subject matter expert and the vendor's development team; producing documents helps reduce knowledge asymmetries between sites and served as a reference point for communicating shared meanings.

Having proposed a model for knowledge exchange in GSD, Kristjansson et al. (P29) proposed a set of practices for shared understanding of specifications based on personalization and codification approaches. While recommending open communication about functionalities between sites (i.e., personalization), they suggested "specification grooming" as a systematic process for review, clarification and revision of approved specifications (i.e., codification) (P29). In our study (P42), however, we presented an artefact-based knowledge sharing system for sharing technical specifications between dispersed teams. We found that the studied organization prepared and shared detailed artifacts (i.e., requirement specification, technical design, user interface design) with offshore as the main means of communication. Identified knowledge sharing system was accompanied with technologies (e.g., Microsoft SharePoint and Email) as well as certain practices (e.g., monitoring update-ness and consistency of artifacts). We found that an artefact-based solution was effective to share project knowledge and make shared understanding in absence of proper social means

for communication between sites. Salger et al. (P27) propose patterns for producing specifications using common terminologies to facilitate sharing and understanding requirement specifications. They (P27) suggest explicitly documenting “business terms” and keeping a reference table in all artefacts for mapping business terms to logical entities.

7.4. Team Cognition Practices

Fostering Transactive Memory. Several studies (P18, P34, P37, P39, P53, P55, P54) referred to promoting transactive memory system (TMS) within GSD teams. Manteli et al. describe (P53) *transactive memory system of a network represents the capability of team members in finding and accessing relevant knowledge in the network*. It is considered as a building block of knowledge sharing (P54) and typically applied to manage the knowledge of “who knows what” in a team. In (P34), Oshri et al. argue that a TMS enables knowledge sharing by creating a catalogue of pointers to knowledge holders and knowledge seekers. They (P34) argue for combining personalization (e.g., joint training, socializations) as well as codification (e.g., directory of expertise) approaches for encoding, storing and decoding knowledge in memory systems. Following with personalization approaches, Kotlarsky et al. (P39) underscored the impact of direct communication between dispersed members to establish and maintain TMS. They further discuss that social practices such as collocated team building was beneficial for key team members to meet, learn about areas of expertise, understand cultural differences and create space for social interactions. On codification practices, however, Kotlarsky et al (P55) emphasise the emergence of codifying “knower” (i.e., codified information about individuals) through means such as comments in source code or version control systems. They (P55) found that that information was helpful to create links between distributed members in terms for identifying and integrating expertise in relevant functional domains and leveraging joint problem-solving. Besides, Avram (P18) observed that provision of technologies such as expertise browsers was helpful to manage knowledge of “who knows what”. Similarly, employing a shared database of skillsets reported to be important by (P37) for visibility of expertise, especially due to high demands for training team members on technical skills.

Verifying Common Understanding between Sites. One of the concerns about sharing knowledge within GSD teams is ensuring the right interpretation of the knowledge shared and having a common understanding. Several of the reviewed studies (P5, P8, P15, P19, P23, P25, P42, P46, P30) proposed different approaches to verifying team members’ understanding of a particular issue or topic. One of the key practices is “Reverse Presentation” (P8, P19, P25, P42, P46) (also called “play back”, “replay session”) to assess the level of understanding by knowledge receivers. This practice is considered as a method of requirement engineering (P25) in which knowledge receivers (e.g., vendor team) present their understanding from specifications to knowledge providers (e.g., client team). This practice has been found to be significantly helpful to identify misunderstandings that could happen due to mistakes in interpretation and/or socio-cultural issues. Reverse presentation is also considered to be a useful mechanism to mitigate issues such as being uncomfortable to ask questions and willing to say “yes” even if there is confusion (P19, P25, P42). Other practices aimed at verifying common understanding include providing feedbacks (P23), running written/ oral tests (P46) and frequent iterations (P15) in this regard. Koch and Sauer (P23) report a formalized practice of providing feedbacks to offshore teams on implemented solutions. The feedback was based on different aspects of software quality (e.g., business requirements, architecture, code style) and found to be effective to clarify implicit conventions and rules. Aman and Nicholson (P5) propose a conceptual framework to diagnose knowledge sharing breakdowns in GSD. Identifying different types of knowledge gaps (e.g., prior knowledge, domain knowledge), they argue that the framework could help dispersed teams to diagnose misunderstandings and filling knowledge gaps. It has also been suggested to utilize questionnaires to systematically capture areas (e.g., domain, IT, organizational process) for knowledge sharing (P30). Vijayakumar et al. (P6) proposed using “Storytelling” technique to simulate informal corridor talks between dispersed teams. They (P6) suggest that this technique is an appropriate instrument for sharing tacit knowledge especially in cross-cultural contexts with a high amount of embedded knowledge.

Improving Team Qualification and Expertise. Our review has identified the practices that aim at raising team qualification to fill knowledge gaps between dispersed team members and facilitate knowledge sharing. We found several studies (P7, P24, P37, P46) that emphasize formal training arrangements by organizations. For instance, one study (P24) found team members were participating in numerous training courses given by the organization; the trainings were considered effective to learn how to access codified knowledge on the

intranet as well as how to interact with team members in organization. Chua and Panb (P46) identified mentoring offshore teams by onshore experts through organizing presentations and providing simulated scenarios for offshore to provide solutions. Some studies (P9, P14) emphasize utilizing highly qualified members in distributed teams. Having high qualification found to be helpful for improving absorptive capacity of dispersed teams (P9) and sharing a higher extent of knowledge (P14).

7.5. Social Practices

Social Ties (Trust and Rapport). The reviewed papers emphasise a significant role for trust and reputation to facilitate knowledge sharing in GSD (P4, P9, P13, P14, P24, P49, P50, P54). It is evident that stronger social ties enable faster exchange of information compared with documentation (P9), and individuals who have higher credibility can transfer a higher volume of knowledge to their trusted network (P14, P50). One study (P24) revealed a tendency of GSD members to seek knowledge through personal networks rather than browsing into codified knowledge. Boden et al (P24) argued that trust (affective and cognitive) significantly influenced the way teammates approach each other for seeking or sharing knowledge. From another perspective, another study (P50) reports that highly trusted individuals in virtual teams are likely to share a higher extent of knowledge to teammates; they could become knowledge centres and emerge as leaders.

Team Cohesion. Team cohesion is defined as “a bond among group members that indicates higher level of mutuality and perception of all members as being a whole” (P35). A shared identity enables a dispersed team’s members to more effectively transfer higher volume of knowledge (P35) and gain a higher level of understanding (P49, P38). Zimmerman and Ravishankar (P49) reported that building social ties and shared identity between dispersed team members significantly helped them to develop *shared contextual understanding*; it accordingly influenced the ability of the dispersed team members to communicate and interpret each others’ messages. Newell et al. (P38) emphasized the need for strategies to move from “them” perspective to “us” in order to experience successful knowledge sharing. The authors suggest that nurturing an organizational culture that demonstrates equal commitment to both offshore and onshore employees is likely to be an enabler of global project teamwork (P38). Sarker et al. (P14) drew attention to the potentials of collectivist vs. individualist national cultures for knowledge seeking and sharing activities in a team.

7.6. Technological Practices

Providing Groupware. While almost all the reviewed studies reported the use of different types of tools (as stated in detail in section 3.3) for supporting knowledge sharing, in this section we consider those studies that particularly referred to tools as a practice. Several studies (P44, P7, P2, P8, P59, P60, P42, P29, P22, P15) reported the provision of knowledge repositories (e.g., Wiki, discussion forums, portals, document management systems) was helpful to codify and centralize organizational knowledge to be accessible for distributed members. Document management systems (e.g., Microsoft SharePoint) have been found to be helpful to manage project documents in distributed settings (P44, P42) by providing features for classification and versioning documents and tracing information. Taweel et al. (P44) reported that introducing a document management system significantly reduced knowledge management issues. The use of Wikis has been reported as an effective means for indexing project knowledge sources between sites (P2) and facilitating individual self-learning (P7). Treude and Storey (P60) explored how Web-based tools such as forums, Wikis, mailing lists and blogs were utilized by large distributed teams to share software development knowledge. They conclude that despite the use of traditional documentation, community portals could be more effective to support knowledge sharing needs of distant teams by providing mechanisms such as “like” an article, leaving feedbacks and involving client in documentation. Several studies referred to the role of different media (e.g., IM, videoconferencing, emails, and screen sharing) for enabling synchronous and asynchronous interaction of distributed teams. The abovementioned tools (i.e., Wikis, information portals, and mailing lists) and media application are also considered Social Software (SoSo) as these tools and application also support social interactions through synchronous and asynchronous communication among team members. Whilst most of the software engineering tools have started to incorporate certain features to social interactions, our review does not explicitly addresses the role of social media in knowledge sharing in GSD. For a detailed discussion on the role of SoSo in GSD, we refer the reader to a systematic mapping study [37] that has specifically address the topic of SoSo in GSD.

Wende et al. (P31) observed that given the socio-cultural issues (e.g., power distance and uncertainty avoidance), IM was the most preferred communication medium used by dispersed team members for multiple purposes such as clarification of questions, socialization as well as sharing status information.

Kotlarsky et al. (P39) discussed that the provision of technologies (e.g., shared databases, shared software development tools, communication media and collaborative tools) in distributed settings help to amplifying knowledge sharing because they provide higher speed and flexibility for sharing knowledge independent of place and time. They argue for the use of Application Sharing Tool (AST) for synchronous knowledge sharing from a distance such as giving presentations and discussing technical issues (e.g. code review, debugging).

Novel Techniques and Solutions. Several of the reviewed studies propose novel techniques and solutions to facilitate knowledge sharing in GSD (P11, P21, P28, P33, P41).

Aranda et al. (P21) proposed a methodology to facilitate choice of appropriate groupware tools (e.g. IM, Email, Wiki, forum, videoconferencing) based on cognitive features. Highlighting the crucial role of groupware tools for communication in distributed environments, they suggest analysing team members' preferences according to their cognitive characteristics to be able to provide the most suitable tool to the team. Presenting preliminary results of employing the methodology in distributed software teams, Aranda et al. (P21) argued that provision of appropriate groupware tools could improve stakeholders' satisfaction with communication and experiencing higher quality of requirement specification shared among distributed members. Averbakh et al. (P41) propose a framework for capturing and sharing experiences of distributed team members. Experiences in this context could vary from typical observations (e.g., difficulties to have an audio conference due to background noise) to strongly sensible and emotional contents. Associating an experience based system with right management, Averbakh et al. (P41) argue that the proposed framework could motivate sharing experiences within a team as it removes confidentiality concerns. Stapel and Schneider (P28) introduced the Flow Mapping technique to address the problem of group awareness. This technique uses Flow notations to visualize information about project participants and their communication paths. The authors (P28) argued that the proposed technique could improve knowledge of team members about each other (e.g., who knows what, who does what) that consequently directs point-to-point communication across sites. In (P33), Mohana and Ramesh present a traceability-based solution to facilitate integration of knowledge in distributed settings. The proposed technical solution collects related artefacts from different tools (e.g., communication media, planning tools, development tools) and facilitates integration and sharing of related knowledge to different team members. Palacio et al. (P11) propose a Knowledge Flow Identification (KoFI) methodology to capture related knowledge from different sources including people, documents and system repositories and provide real-time awareness about the status of collaborators to each other. Information such as team members' presence and identity, artefacts in use, people and resource location, associated projects, plans and history logs are retrieved through technology and provided to distributed team members to overcome physical distance that can prevent the knowledge flow among collaborators. The proposed methodology was prototyped for interactive Instant Messaging that supports a real-time interaction of dispersed team members in GSD.

8. Limitations

One of the threats to internal validity of results from this SLR could be related to comprehensiveness and coverage of the relevant studies. This risk could be more likely in the context of global software development. As stated by Smite and colleagues [38] many variations and settings of developing software with globally distributed teams have resulted in introducing a large number of new terms, which cause obstacles to discover related work through systematic approaches. If the relevant studies do not use the standard terminologies in title and abstract, they inhibit the comprehensive coverage of SLR from published works on this topic [38]. We should note that we ran our search string on the Scopus indexing system to ensure consistency of automatic search in terms of structure and targeting metadata (as elaborated in section 2.2.2). While we are confident about the coverage of Scopus on a large number of venues in software engineering and computer science [23, 24], there could be a risk involved due to the use of a single indexing system. We have mitigated such a risk of systematic omission of a large number of studies, by consulting with sample SLRs (e.g. [14, 37, 39]) for building our search string and also applying snowballing (i.e., manual search on references of selected studies), however, limitations in this regard should be taken into consideration.

The other validity threat could be related to selection, analysis and synthesis of the extracted data being biased by the interpretation of the researcher. Inclusion/ exclusion of studies has passed through accurate

selection, on-going internal discussion and crosschecking between the authors of the SLR. Furthermore, collecting data by two extractors was also helpful to minimize any risk of researchers' own bias.

9. Comparison of our Work with an Existing SLR

As previously stated, this SLR has some overlaps with a study that used a mixed methods (i.e., SLR and Interviews) approach to explore the topic of knowledge transfer in GSD [19]. Whilst the high level objective of both the studies appears to be similar, i.e., understanding the knowledge sharing challenges and practices in GSD, these studies differ with each other in the use of methodological approaches and how the used methodologies were applied leading to several differences. In followings we state the major dissimilarities:

- The search string used in [19] has been scoped to “*knowledge transfer*” and the synonyms. For unknown reasons, researchers did not include well-known keywords such as “*knowledge management*” and “*knowledge sharing*”. The choice of keywords might have retrieved the studies that focused on knowledge transfer only. We argue that not using the well-known keywords would have limited the search string's ability to retrieve the most of the papers related to knowledge sharing in GSD.
- There is almost 3.5 years difference between running search string by [19] (i.e., March 2011) and us (i.e., September 2014). Given the evidence [19], our review incorporate more recent studies published after March 2011.
- Our study followed different inclusion/exclusion criteria. We aimed at reviewing empirical studies presenting findings only from primary data, to fulfill our third research question. However, the previous SLR did not focus on the empirical aspects of the reviewed studies; the previous review used only automatic search, we applied automated searches and snowballing for finding the relevant papers.
- Due to the abovementioned reasons, there is a significant difference in the papers reviewed by both SLRs. Out of 61 papers in our SLR and 35 papers in [19], there were only 11 common papers.

The objective of our review was not to quantitatively assess the identified challenges and practices of Nidhra et al. [19], rather we aimed at analysing the identified primary studies independently based on our own understanding from the topic and providing readers a different set of insights into this important topic. Our approach clearly led to differences in coding schemes, level of details and emergence of categories, which means one-to-one comparison of the identified challenges and practices was not be reasonable for drawing any conclusions. Exemplifying some of the major differences, our review underscored the popular practice of “boundary spanning” (i.e., found by 10 studies), which was not reflected in the previous review. Our findings have highlighted the importance and popularity of “temporary collocation” for supporting knowledge sharing in GSD teams (i.e., found by 17 studies). We elaborated on the importance of visits for building social relations, dissemination of knowledge and effective team learning. The previous SLR did not cover this important practice, i.e., cited by 3 studies under different codes.

10. Discussion

Knowledge sharing is an integral part of knowledge management, which is considered a critical support mechanism for successful completion of knowledge intensive activities like software development. It has become clear that distribution of software development teams, a paradigm called GSD, creates different sorts of knowledge sharing challenges that need to be addressed by devising and implementing appropriate practice. The goal of this research was to systematically review and synthesize the literature published on the topic in an attempt to help improve the understanding of the contextual aspects of the published studies and classify the identified knowledge sharing challenges and practices in GSD.

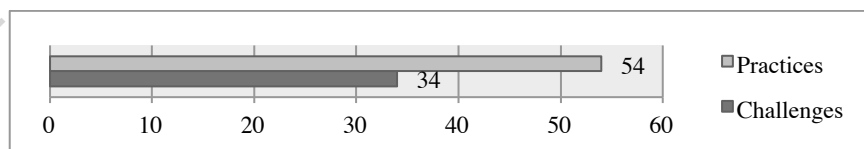


Figure 9 – Distribution of Primary Studies based on Knowledge Sharing Challenges and Practices

Figure 9 shows that 61% of primary studies (54 out of 61) reported knowledge sharing practices while 39% (34 out of 61) reported challenges of sharing knowledge in GSD teams. There were several papers that reported challenges as well as practices. Figure 10 shows among the 10 papers published in 2014, all studies

referred to practices while only 5 studies presented challenges. From Figure 9 and 10, we observe that there is relatively higher tendency among researchers presenting practices to support knowledge sharing in GSD. This can be a clear indication that there is a significant realization of knowledge sharing challenges; hence more practical solutions are getting introduced. However, it does not necessarily mean that knowledge-sharing challenges in GSD are disappearing as they have been continuously reported over the last decade.

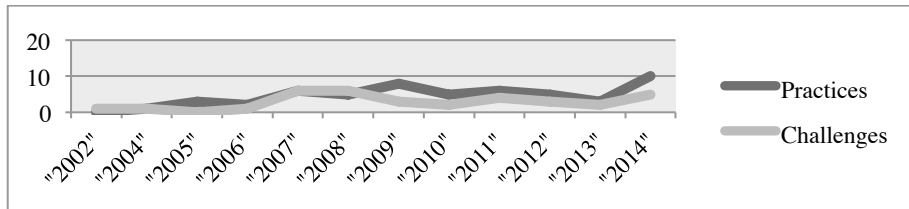


Figure 10 – Trend of Reporting Knowledge Sharing Challenges and Practices over Years

Based on our analysis, we have identified 6 main themes under which the challenges and practices of knowledge sharing in GSD can be classified: management, team structure, work processes, team cognition, social attributes and technology and techniques (i.e., elaborated in sections 6, 7). Figure 11 presents distribution of primary studies based on these themes from the perspectives of challenges and practices. As it can be seen, “Work Processes” with 30 studies include the most and “Social attributes” and “Technology/Techniques” each with 20 studies include the least number of studies. The other categories did not reflect significant difference in number of included papers: management (26 studies), team cognition (24 studies) and team structure (22 studies).

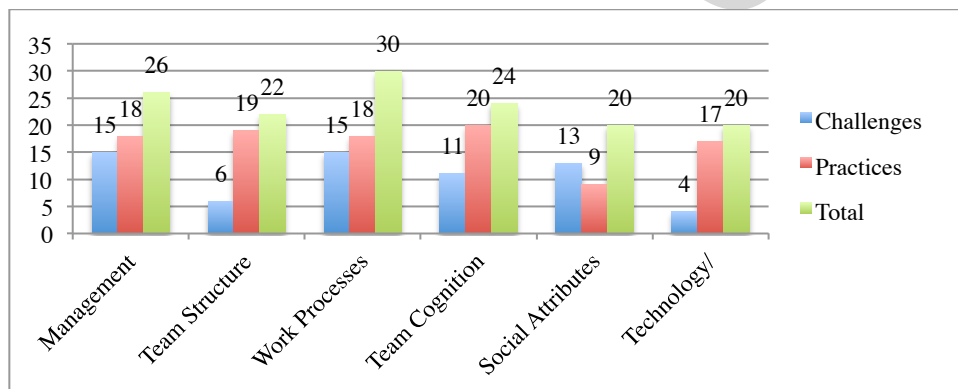


Figure 11- Distribution of Papers based on Themes of Knowledge Sharing Challenges and Practices

Table 9 summarizes the most frequently reported challenges and practices in the reviewed papers. Among all the identified challenges explained in section 6, the cost of knowledge sharing, contextual difference, and lack of openness have been reported more frequently than any other challenges; employee turnover and documentation problems have also been reported by several studies. Decreasing the cost of software development is a well-known motivation for utilizing human resources from other geographies; however, GSD can have hidden cost to projects that can negatively affect the expected economic benefits. Our review underscored that sharing knowledge in GSD can be associated with costs (e.g., for travelling between sites, maintaining detailed documentation) that may not be expected beforehand, which impact project plans.

Our review also suggests that one of the major challenges of GSD teams is to deal with the context specific tacit knowledge. Many studies supported the commonly known anecdotal view that lack of understanding the domain at offshore can cause misinterpretation of requirements. It has been discussed that business domain is quite difficult to share with remote sites; domain knowledge is highly tacit and usually embedded in a particular context, which hardens identifying the details that need to be transferred (P8, P5, P30, P46). One of the key reasons of this situation is that socio-cultural features of teams can inhibit the abilities of the knowledge sender to assess the understanding of the recipient to provide the required clarification to resolve doubts when required. It is evident that whilst knowledge diversity can trigger knowledge sharing and result in creativity and innovation [40], it can also impede the knowledge flow, cause misunderstanding and result in failures to stimulate innovation [41]. In fact, both knowledge diversity and common knowledge need to

reinforce each other to enable knowledge sharing [10, 11]. De Souza and Awazu [41] discuss that common knowledge in organizations provides a common frame of reference, which facilitates communication and sense-making from exchanged knowledge.

Table 9 – The Most Frequent Challenges and Practices identified in this SLR

	Factors	# (n=61)
Challenges	Cost of Knowledge Sharing	7
	Contextual Difference	7
	Lack of Openness	7
	Employee Turnover	6
	Documentation Problem (missing, poor, outdated)	6
Practices	Temporary Collocation	17
	Frequent Communication	12
	Providing Groupware	12
	Using Boundary Spanning Roles	10
	Verifying Understanding between Sites	10
	Fostering Transactive Memory	7
	Social Ties (Trust and Rapport)	7

Among the identified practices discussed in Section 7, our findings have identified that the “temporary collocation” as the most popular practice for sharing knowledge in GSD reported in 17 studies. While visiting remote sites is a widely known practice in GSD, we assert that the significance of this practice deserves more exploration and in-depth analysis to investigate how temporary collocation contributes to knowledge sharing. For example, similar approaches to (P1) for providing detailed information such as frequency of visits, duration, participants and organized activities during visits could guide practitioners to effectively employ the practice. In addition to mobilizing individuals between sites and benefiting from face-to-face communication, our study has also observed the popularity of practices to support knowledge sharing from a distance by providing groupware tools (i.e., referred by 12 studies) for enabling frequent communication between remote team members (i.e., referred by 12 studies). Other knowledge sharing practices that were reported by several studies include utilizing boundary-spanning roles (10 studies), verifying understanding between sites (10 studies), fostering transactive memory (7 studies) and building social ties (7 studies). Having looked at the list of the most popular practices, we can conclude that an increasing number of companies tend to leverage social potentials of teams for facilitating knowledge sharing in GSD.

Our review identified only a few knowledge sharing challenges related to technological issues (i.e., reported in 4 studies). Some of the identified challenges were related to inappropriate application or use of tools (e.g., not populating Wikis). A significant number of the identified knowledge sharing practices emphasise the role of providing suitable technologies to support knowledge sharing needs of distributed teams. These practices advocate the use of a variety of communication media and groupware tools such as email, audio/video conferencing, IM, issue-tracking system, shared network drive, Wiki and discussion forums.

Our review has revealed that the emerging trend of capturing and analysing meta-knowledge in distributed teams (P11, P28, P33) for getting information about knowledge sources (i.e., Actors and Artefacts) and their interactions to organize memory of distributed teams about areas such as distribution of expertise, identity and availability of dispersed members, contribution of members to project artefacts and involvement of team members in different project tasks. Another emerging trend is to develop tools that can help capture and organize the chunks of knowledge that usually remain unshared between distributed teams (P11, P28, P33, P41). Such tools are expected to support integration of knowledge scattered across different technologies (e.g., IM, emails, project planning, and source code comments). We can conclude that there are a large number of technologies available for supporting knowledge sharing needs of GSD teams to a large extent.

We were specifically interested in identifying the contextual attributes that characterised the reviewed studies. There has been an increasing realisation that software development challenges, lessons learnt and best practices need to be explored and understood along with their respective contexts [42]. There are several attempts by researchers to formulate the context of software development work [42-44] to support the exploration of the impact of contextual attributes on different aspects of software development such as

success/failure of projects, and assessing usefulness of a practice. There is an increasing advocacy for systematic literature reviews to avoid drawing abstract conclusions based on de-contextualized findings [42]. We decided to extract and analyse a large number of contextual attributes from the reviewed studies. We assert that organizational contexts play an important role in creating and addressing knowledge sharing challenges; hence, they should be taken into consideration in any study. For instance, given the differences between large enterprises and small/medium-sized enterprises (SMEs) in available budget, number of human resources, maturity of processes and practices, they are likely to experience different nature of challenges and accordingly employ different mitigation strategies. Dingsøyr and Royvik [45] discuss that large organizations can afford resources required for knowledge management; that is why a codification strategy, even though being expensive in terms of effort and required infrastructure, are normally implemented by large enterprises. However, limited resources of SMEs may not permit heavy investment on codification [45] and demand lighter strategies based on personalization [41].

Our approach to examine contextual settings of primary studies covered two dimensions: methodological and substantive context (i.e. context that individuals face) [42].

It is widely known that research methods such as ethnography and case study offer strong tools to explore the context of a phenomenon under study [46, 47]. These methods can provide a researcher with more opportunities to interact with participants in their real-world environment, observe dynamics and collect more qualitative data, which can enable a researcher to understand the circumstances around the studied phenomenon. Research methods such as controlled experiment provide fewer possibilities to study contextual aspects. A majority of the reviewed primary studies were conducted using the case study research method (i.e., 43 studies); that means these studies could have extensively explored the contextual settings. We have considered the methodological aspects of the reviewed studies in our quality check process and excluded the papers that were poor in reporting the methodological settings. We did not include the studies that solely probed causal relations between abstracted concepts through totally de-contextualized approaches, e.g., conducting large-scale survey with practitioners from different organizations.

We faced several difficulties in identifying and extracting the contextual information as there was a general lack of contextual information in most of the reviewed studies. Based on our internal discussion, we initially identified several contextual attributes that were expected to have the potential to impact knowledge sharing challenges and practices in GSD such as business domain, work distribution strategy between sites, size of team, type of project (e.g., maintenance and greenfield), software methodology (e.g., agile) and process maturity level. However, a general lack of information about the abovementioned contextual aspects, we decided to focus on high-level attributes that are typically reported in the GSD literature: number and location of sites, collaboration model (e.g., outsourcing and insourcing) and size of organization. Apparently, not all the studies provided this information either, which resulted in categorizing them under the “unclear” category. We observe that even though GSD research predominantly utilizes context-wise rich methodologies that are suitable to capture a study’s context, GSD researchers fail to report sufficient contextual information. We also note that this situation gets even worse when a paper reports the findings from more than one case as it becomes difficult to differentiate the findings from one case to another case.

Whilst it is difficult to draw a definitive conclusion about the context specific nature of the identified practices, we have observed that some practices were more popular in certain settings. For instance, flexible communication structures and joint work between sites were mainly found in studies of SMEs (i.e. <100 employees), while practices such as heavy documentation and providing knowledge repositories were mainly implemented in large enterprises (i.e. > 1000 employees) for supporting knowledge sharing in GSD.

11. Conclusions and Implications

This paper reports our research effort aimed at systematically reviewing and analyzing knowledge sharing challenges and practices in GSD. Based on a rigorous analysis and systematic synthesis of the 61 papers reporting empirical studies of knowledge sharing challenges and practices in GSD, we can conclude:

- (1) A majority of the reviewed primary studies (i.e., 43 studies) used the *case study* research method. A large majority of the studies were conducted in *industrial* (i.e., 54 studies) rather than *academic* (i.e., 7 studies) settings, while *Interview* was the most frequent data collection approach (49 studies), followed by *documentations/archival records* (29 studies).

- (2) The results of this SLR show that the *offshore-outsourcing* collaboration model was dominant in GSD and large organizations (i.e., >1000 employees) were so popular to carry out the research on GSD.
- (3) The identified knowledge sharing challenges and practices in GSD can be classified in 6 main themes: management, team structure, work processes/ practices, team cognition, social attributes and technology, in which the *cost of knowledge sharing*, *contextual difference*, and *lack of openness* were more frequent challenges for knowledge sharing. Whilst *temporary collocation*, *frequent communication* and *providing groupware* were popular practices to support knowledge sharing.

The results of this SLR can have several implications for researchers, practitioners, and tool developers in the area of GSD. For researchers, this review has identified a number of potentially researchable topics. For example, (i) as reported in this SLR, a majority of the included studies did not report sufficiently the contextual information; we strongly suggest that more attention should be paid to reporting contextual factors by researchers, which helps improve the quality and credibility of the reported knowledge sharing challenges and practices; (ii) since small/medium-sized enterprises (SMEs) have more freedom and flexibility for experimenting in innovative practices, hence, researchers can benefit from such context to systematically devise, implement and evaluate new generations of practices and tools for supporting knowledge sharing in GSD; (iii) a vast majority of the reviewed studies did not consider the cost and benefit factors of the proposed knowledge sharing challenges and practices in GSD. Hence, we assert that there is an important and urgent need of sufficient research work to be conducted for cost-benefit analysis of the proposed challenges and practices.

The findings from this review also provide potentially useful insights for GSD practitioners and tools developers. (i) Practitioners can use the findings of this SLR to gain a better understanding of the available knowledge sharing practices and tools and their suitability for different contexts in a way that they can apply practices to address their knowledge sharing challenges; For example, whilst it is important for practitioners to know a lack of trust between dispersed members inhibits sharing knowledge, they should be provided with practical solutions to build and sustain trust in GSD. (ii) the findings from our review also indicate that disparity in technical and domain knowledge of distributed team members can lead to several types of knowledge sharing challenges; hence, companies need to pay significant attention to bridge the technical and business domain knowledge gaps in their GSD teams' members located at different places; (iii) this review has found a few recent meta-level tools that utilize existing technologies, leverage collective memory of distributed members about available expertise and promote sharing and seeking knowledge at the level of individuals. We suggest that researchers, practitioners, and tool developers can cooperate closely to enhance meta-level tools for supporting knowledge sharing in GSD settings.

The topic of this review falls at the intersection of Software Engineering and Information Management disciplines. Traditionally Information Management researchers have studied knowledge management (and knowledge sharing) more frequently than software engineering researchers. However, software engineering researchers and practitioners have been paying an increasing amount of attention to knowledge management in software engineering, particularly in the context of GSD. Hence, the findings from this review are potentially beneficial to both software engineering as well as information management communities. This review provides information about knowledge sharing from software engineering perspective and highlights the criticality of the topic when software development teams are globally distributed. We assert that exploring knowledge management in outsourcing business models is a very familiar topic to the Information Management community, however, this review will help them to know why and how knowledge sharing is increasingly becoming important area of research and practice in the context of distributed software engineering. Software development involves highly intellectual activities in which sharing business domain and technical knowledge among team members is crucial for successfully delivering software product. It significantly differs from producing other products (e.g., assembling electric devices) by remote sites in terms of difficulties and criticality of sharing knowledge among geographically dispersed team members. Therefore, it is important for both Information Management and Software Engineering communities to develop an in-depth understanding of knowledge sharing needs and challenges of distributed software teams and providing solutions that are appropriated for GSD context. This review is expected to help them to gain such an understanding.

In this SLR, for the sake of clarity and coherency of the results, we decided to include only those papers that report knowledge sharing in the context of proprietary GSD. Nevertheless, it would be quite useful to extend

this review to perform a comparative study of knowledge sharing challenges and practices in Open Source software development viz-a-viz proprietary GSD. We can assume that such a comparison would provide useful information about similarities and differences of knowledge sharing challenges and practices in two different but similar arrangements of developing software. Whilst open source communities usually comprised of individuals that are distributed across the World, they are characterized as highly cooperative teams that are willing to participate in software development because of their personal interests and learning objectives. The team atmosphere and social relationships in open source communities vary from some of the proprietary GSD models (e.g., outsourcing) where there is a risk of losing competency and competitiveness in the market by sharing knowledge with external companies. Our findings in this SLR revealed six high-level categories (i.e., Management, Team Structure, Work Processes, Team Cognition, Social Attributes and Technology) around which knowledge sharing challenges and practices of GSD teams could be realized. We suggest that these categories can be used as a baseline for conducting a comparative study of knowledge sharing challenges and practices in Open Source software development and proprietary GSD.

Acknowledgements

This research has been partially funded by the project “NexGSD” through grant number #10-092313.

References

1. Herbsleb, J.D., et al., *An empirical study of global software development: distance and speed*, in *Proceedings of the 23rd International Conference on Software Engineering*. 2001, IEEE Computer Society: Toronto, Ontario, Canada. p. 81-90.
2. Damian, D. and D. Moitra, *Guest Editors' Introduction: Global Software Development: How Far Have We Come?* Software, IEEE, 2006. **23**(5): p. 17-19.
3. Bhat, J.M., G. Mayank, and S.N. Murthy, *Overcoming Requirements Engineering Challenges: Lessons from Offshore Outsourcing*. Software, IEEE, 2006. **23**(5).
4. Herbsleb, J. and D. Moitra, *Global software development*. Software, IEEE, 2002. **18**(2): p. 16-20.
5. Grinter, R.E., J.D. Herbsleb, and D.E. Perry, *The geography of coordination: dealing with distance in R&D work*, in *Proceedings of the international ACM SIGGROUP conference on Supporting group work*. 1999, ACM: Phoenix, Arizona, United States. p. 306-315.
6. Ågerfalk, P., et al., *A framework for considering opportunities and threats in distributed software development*, in *Proceeding of International Workshop on Distributed Software Development (DiSD)*. 2005, Austrian Computer Society.
7. Carmel, E. and R. Agarwal, *Tactical approaches for alleviating distance in global software development*. Software, IEEE, 2001. **18**(2).
8. Kotlarsky, J. and I. Oshri, *Social ties, knowledge sharing and successful collaboration in globally distributed system development projects*. European Journal of Information Systems, 2005. **14**(1).
9. Khan, S.U., M. Niazi, and R. Ahmad. *Critical Success Factors for Offshore Software Development Outsourcing Vendors: A Systematic Literature Review*. in *Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on*. 2009.
10. Choo, C.W. and R.C.D. de Alvarenga Neto, *Beyond the ba: managing enabling contexts in knowledge organizations*. Journal of Knowledge Management, 2010. **14**(4): p. 592-610.
11. Santos, V., A. Goldman, and C.R. de Souza, *Fostering effective inter-team knowledge sharing in agile software development*. Empirical Software Engineering, 2014: p. 1-46.
12. Cummings, J.N., *Work groups, structural diversity, and knowledge sharing in a global organization*. Management science, 2004. **50**(3): p. 352-364.
13. Ebert, C. and J.D. Man, *Effectively utilizing project, product and process knowledge*. Information and Software Technology, 2008. **50**(6): p. 579-594.
14. Bjornson, F.O. and T. Dingsoyr, *Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used*. Information and Software Technology, 2008. **50**(11): p. 1055-1068.
15. Boden, A., et al., *Knowledge sharing practices and the impact of cultural factors: Reflections on two case studies of offshoring in SME*. Journal of software: Evolution and Process, 2012. **24**(2): p. 139-152.
16. Noll, J., S. Beecham, and I. Richardson, *Global software development and collaboration: barriers and solutions*. ACM Inroads, 2011. **1**(3): p. 66-78.
17. Hinds, P. and C. McGrath, *Structures that work: social structure, work structure and coordination ease in geographically distributed teams*, in *CSCW*. 2006, ACM: Canada.

18. Coleman, J.S., *Social capital in the creation of human capital*. American Journal of Sociology, 1988.
19. Nidhra, S., et al., *Knowledge transfer challenges and mitigation strategies in global software development -A systematic literature review and industrial validation*. International Journal of Information Management, 2013. **33**(2): p. 333-355.
20. Kitchenham, B.A., T. Dyba, and M. Jorgensen. *Evidence-based software engineering*. in *International Conference on Software Engineering(ICSE)*. 2004.
21. Kitchenham, B. and S. Charters, *Guidelines for performing systematic literature reviews in software engineering*. Engineering, 2007. **2**(EBSE 2007-001).
22. Zhang, H., M.A. Babar, and P. Tell, *Identifying relevant studies in software engineering*. Information and Software Technology, 2011. **53**(6): p. 625-637.
23. Kitchenham, B., et al., *Systematic literature reviews in software engineering—a tertiary study*. Information and Software Technology, 2010. **52**(8): p. 792-805.
24. Daneva, M., et al., *Empirical research methodologies and studies in Requirements Engineering: How far did we come?* Journal of Systems and Software, 2014. **95**: p. 1-9.
25. Burnham, J.F., *Scopus database: a review*. Biomedical digital libraries, 2006. **3**(1): p. 1.
26. Budgen, D., et al. *Using mapping studies in software engineering*. in *Proceedings of PPiG*. 2008.
27. Braun, V. and V. Clarke, *Using thematic analysis in psychology*. Qualitative Research in Psychology, 2006. **3**(2).
28. Niinimäki, T., et al. *Reflecting the Choice and Usage of Communication Tools in GSD Projects with Media Synchronicity Theory*. in *International Conference on Global Software Engineering (ICGSE)*. 2010: IEEE.
29. Tell, P. and M. Ali Babar, *A Systematic Mapping Study of Tools for Distributed Software Development Teams*. 2012.
30. Whitehead, J., *Collaboration in software engineering: a roadmap*. Future of software engineering, 2007: p. 214-225.
31. Chen, L. and M.A. Babar, *A systematic review of evaluation of variability management approaches in software product lines*. Information and Software Technology, 2011. **53**(4): p. 344-362.
32. Yin, R., *Case study research: Design and methods*. 2003: Sage Publications, Inc.
33. Easterbrook, S., et al., *Selecting empirical methods for software engineering research*, in *Guide to advanced empirical software engineering*. 2008, Springer. p. 285-311.
34. Espinosa, J.A., et al., *Team knowledge and coordination in geographically distributed software development*. Journal of Management Information Systems, 2007. **24**(1): p. 135-169.
35. Hinds, P.J. and S.P. Weisband, *Knowledge sharing and shared understanding in virtual teams*. Virtual teams that work: Creating conditions for virtual team effectiveness, 2003: p. 21-36.
36. Sole, D. and A. Edmondson, *Situated knowledge and learning in dispersed teams*. British journal of management, 2002. **13**(S2): p. S17-S34.
37. Giuffrida, R. and Y. Dittrich, *Empirical studies on the use of social software in global software development - A systematic mapping study*. Information and Software Technology, 2013. **55**(7): p. 1143-1164.
38. Šmite, D., et al., *An empirically based terminology and taxonomy for global software engineering*. Empirical Software Engineering, 2012: p. 1-49.
39. Šmite, D., et al., *Empirical evidence in global software engineering: a systematic review*. Empirical software engineering, 2010. **15**(1): p. 91-118.
40. Orlikowski, W.J., *Knowing in Practice: Enacting a Collective Capability in Distributed Organizing*. Organization science, 2002. **13**(3): p. 249-273.
41. Desouza, K.C. and Y. Awazu, *Knowledge management at SMEs: five peculiarities*. Journal of knowledge management, 2006. **10**(1): p. 32-43.
42. Dybå, T., D.I. Sjøberg, and D.S. Cruzes. *What works for whom, where, when, and why?: on the role of context in empirical software engineering*. in *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement*. 2012: ACM.
43. Petersen, K. and C. Wohlin, *Context in industrial software engineering research*, in *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*. 2009, IEEE Computer Society. p. 401-404.
44. Kirk, D. and S.G. MacDonell. *Investigating a conceptual construct for software context*. in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. 2014: ACM.
45. Dingsøyr, T. and E. Røyrvik. *An empirical study of an informal knowledge repository in a medium-sized software consulting company*. in *Proceedings of the 25th international conference on Software engineering*. 2003: IEEE Computer Society.
46. Runeson, P. and M. Höst, *Guidelines for conducting and reporting case study research in software engineering*. Empirical Software Engineering, 2009. **14**(2): p. 131-164.
47. Yin, R.K., *Applications of case study research*. 3rd ed. 2012: SAGE Publications, Incorporated.

Appendix A. Primary Studies

Table 10 - Primary Studies

ID	Title	Author(s)	Venue	Year
P1	The four 'W' of Face-to-Face - Suggesting an Enriched Perspective on Nearshoring Relationship Management	A. von Stetten, D. Beimborn, T. Weizel	Conference on Computers and People Research	2014
P2	Managing Knowledge in Global Software Development Projects	T. Dingsøy, D. Smite	IT Professional	2014
P3	The knowledge-bridging process in software offshoring from Japan to Vietnam	N.T. Huong, U. Katsuhiro, D.H. Chi	The Electronic Journal of Information Systems in Developing Countries	2014
P4	Spatial Knowledge Creation and Sharing Activities in a Distributed Agile Project	M.A. Razzak, R. Ahmed, D. Smite	International Conference on Global Software Development Workshops	2013
P5	MIND THE GAP! Understanding Knowledge in Global Software Teams	A. Aman, B. Nicholson	International Federation for Information Processing	2008
P6	Storytelling – a Method to Start Knowledge Transfer in Offshore Software Development Teams – Research in Progress Paper	V. Vijayakumar, R. Gey, E. Wende	European Conference on Knowledge Management	2008
P7	Knowledge Management in Distributed Agile Software Development	S. Dorairaj, J. Noble, P. Malik	Agile Conference	2012
P8	Knowledge Transfer in Offshore Outsourcing Software Development Projects: An Analysis of the Challenges and Solutions from German Clients	S. Betz, A. Oberweis, R. Stephan	Expert Systems	2014
P9	Knowledge Sharing Barriers in Global Teams	M. Wendling, M. Oliveira, A.C.G. Macada	Journal of Systems and Information Technology	2013
P10	An Empirical Study on Improving Shared Understanding of Requirements in GSD	M. Humayun, C. Gang	International Journal of Software Engineering and Its Applications	2013
P11	Knowledge Flow as Facilitator for Getting into Collaboration in Distributed Software Development	R. R. Palacio, A. L. Morán, A. Vizcaíno, V.M. González	Hawaii International Conference on System Sciences	2011
P12	Operational and Strategic Learning in Global Software Development	A. Boden, B. Nett, V. Wulf	IEEE Software	2010
P13	Managing Offshore Outsourcing of Knowledge-intensive Projects A People Centric Approach	M. Jensen, S. Menon, L.E Mangset, V. Dalberg	International Conference on Global Software Engineering	2007
P14	Knowledge Transfer in Virtual Systems Development Teams: An Exploratory Study of Four Key Enablers	S. Sarker, S. Sarker, D.B. Nicholson, K. D. Joshi	IEEE Transactions on Professional Communication	2005
P15	Can Distributed Software Development Be Agile?	B. Ramesh, L. CAO, K. Mohan, P. XU	Communications of the ACM	2006
P16	Collaboration Patterns and the Impact of Distance on Awareness in Requirements-Centred Social Networks	D. Damian, S. Marczak, I. Kwan	IEEE International Requirements Engineering	2007
P17	Globally Distributed Component-based Software Development: An Exploratory Study of Knowledge Management and Work Division	J. Kotlarsky, I. Oshri, J. van Hillegersberg, K. Kumar	Journal of Information Technology	2007

P18	Of Deadlocks and Peopleware - Collaborative Work Practices in Global Software Development	G. Avram	International Conference on Global Software Engineering	2007
P19	Breaching the Knowledge Transfer Blockade in IT Offshore Outsourcing Projects – A Case from the Financial Services Industry	R. Gregory, R. Beck, M. Prifling	Hawaii International Conference on System Sciences	2009
P20	Workgroup Structures in Offshore Software Development Projects: A Vendor Case Study	A. Mathrani, D. Parsons, R. Stockdale	International Enterprise Distributed Object Computing Workshop	2009
P21	Which Groupware Tool is the Most Suitable for This Group?	G. N. Aranda, A. Vizcaíno, M. Piattini	IEEE International Conference on Global Software Engineering	2009
P22	Use of Collaborative Technologies and Knowledge Sharing in Co-located and Distributed Teams: Towards the 24-h Knowledge Factory	A. Gupta, E. Mattarelli, S. Seshasai, J. Broschak	Journal of Strategic Information Systems	2009
P23	A Task-driven Approach on Agile Knowledge Transfer	J. Koch, J. Sauer	Agility across Time and Space	2010
P24	An Understanding of the Role of Trust in Knowledge Seeking and Acceptance Practices in Distributed Development Teams	B. Al-Ani, H. Wilensky, D. Redmiles, E. Simmons	International Conference on Global Software Engineering	2011
P25	Reverse Presentations A Client-Driven Method for Requirements Engineering in Offshore Software Development	M. Wiener, R. Stephan	Journal of Business and Information Systems Engineering	2010
P26	Knowledge Sharing Practices and the Impact of Cultural Factors: Reflections on Two Case Studies of Offshoring in SME	A. Boden, G. Avram, L. Bannon, V. Wulf	Journal of software: Evolution and Process	2012
P27	Towards Specification Patterns for Global Software Development Projects	F. Salger, J. Englert, G. Engels	International Conference on the Quality of Information and Communications Technology	2010
P28	Managing Knowledge on Communication and Information Flow in Global Software Projects	K. Stapel, K. Schneider	Expert Systems	2012
P29	Integration by Communication: Knowledge Exchange in Global Outsourcing of Product software Development	B. Kristjansson, R. Helms, S. Brinkkemper	Expert Systems	2012
P30	Knowledge Transfer Planning and Execution in Offshore Outsourcing: an Applied Approach	S. Madsen, K. Bødker, T. Tøth	Information Systems Frontiers	2014
P31	Exploring the Media Mix during IT-Offshore Project	E. Wende, G. Schwabe, T. Philip	Global Sourcing of Information Technology and Business Processes	2010
P32	Embedded Knowledge and Offshore Software Development	B. Nicholson, S. Sahay	Information and Organization	2004
P33	Traceability-based Knowledge Integration in Group Decision and Negotiation Activities	K. Mohana, B. Ramesh	Decision Support Systems	2007
P34	Transactive Memory and the Transfer of Knowledge between Onsite and Offshore IT Outsourcing Teams	I. Oshri, J. Kotlarsky, P. van Fenema	Information Systems Outsourcing: Enduring Themes, Global Challenges, and Process Opportunities	2009
P35	Seeing Remote Team Members as Leaders: A Study of US-Scandinavian Teams	S. Sarker, S. Sarker, C. Schneider	IEEE Transactions on Professional Communication	2009
P36	Offshore-Onsite Subgroup Dynamics in Globally Distributed Teams	E. Mattarelli, A. Gupta	Information Technology & People	2009
P37	Uncovering the Reality Within Virtual Software Teams	V. Casey, I. Richardson	International workshop on Global software development	2006

			for the practitioner	
P38	An Analysis of Trust Among Globally Distributed Work Teams in an Organizational Setting	S. Newell, G. David, D. Chand	Knowledge and Process Management	2007
P39	Developing a Knowledge-based Perspective on Coordination: The Case of Global Software Projects	J. Kotlarsky, P. C. van Fenema, L.P. Willcocks	Information and Management	2008
P40	Architectural Knowledge Management Practices in Agile Global Software Development	V. Clerc, P. Lago, H. van Vliet	International Conference on Global Software Engineering Workshops	2011
P41	An Experience Base with Rights Management for Global Software Engineering	A. Averbakh, E. Knauss, O. Liskin	International Conference on Knowledge Management and Knowledge Technologies	2011
P42	Knowledge Sharing for Common Understanding of Technical Specifications Through Artifactual Culture	M. Zahedi, M. Ali Babar	International Conference on Evaluation and Assessment in Software Engineering	2014
P43	Towards an Understanding of Enabling Process Knowing in Global software Development: A case study	M. Zahedi, M. Ali Babar	International Conference on Software and System Process	2014
P44	Communication, Knowledge and Co-ordination Management in Globally Distributed Software Development: Informed by a scientific Software Engineering Case Study	A. Taweel, B. Delaney, T.N. Arvanitis, L. Zhao	IEEE International Conference on Global Software Engineering	2009
P45	A US Client's Learning from Outsourcing IT Work Offshore	J.W. Rottman, M.C. Lacity	Information Systems Frontiers	2008
P46	Knowledge Transfer and Organizational Learning in IS Offshore Sourcing	A.L. Chuua, S. L. Panb	OMEGA : International Journal of Management Science	2008
P47	Piloting Knowledge Transfer in IT/IS Outsourcing Relationship Towards Sustainable Knowledge Transfer Process: Learnings from Swiss Financial Institution	M. Bugajska	Americas Conference on Information Systems	2007
P48	Explaining Variations in Client Extra Costs between Software Projects Offshored to India	J. Dibbern, J. Winkler, A. Heinzl	MIS Quarterly - Management Information Systems	2008
P49	Knowledge transfer in IT offshoring relationships: the roles of social capital, efficacy and outcome expectations	A. Zimmermann, M.N. Ravishankar	Information Systems Journal	2014
P50	Path to "Stardom" in Globally Distributed Hybrid Teams - An Examination of a Knowledge-Centered Perspective using Social Network Analysis	S. Sarker, S. Sarker, S. Kirkeby, S. Chakraborty	Decision Sciences	2011
P51	Knowledge Management Initiatives in Offshore Software Development - Vendors' Perspectives	A. Mathrani, D. Parsons, S. Mathrani	Journal of Universal Computer Science	2012
P52	Managing Meta-learning in Offshore Software Development Environments	A. Mathrani, D. Parsons	Journal of Management Development	2012
P53	The Effect of Governance on Global Software Development: An Empirical Research in Transactive Memory	C. Manteli, B. van den Hooff, H. van Vliet	Information and Software Technology	2014
P54	Social Ties, Knowledge Sharing and Successful Collaboration in Globally Distributed System Development Projects	J. Kotlarsky, I. Oshri,	European Journal of Information Systems	2005
P55	Coordinating Expertise Across Knowledge Boundaries in Offshore-Outsourcing Projects: The Role of Codification	J. Kotlarsky, H. Scarbrough, I. Oshri	MIS Quarterly - Management Information Systems	2014
P56	Awareness in the Wild: Why Communication Breakdowns Occur	D. Damian, L. Izquierdo, J. Singer, I. Kwan	International Conference on Global Software Engineering	2007

P57	Small Firms and Offshore Software Outsourcing High Transaction Costs and Their Mitigation	E. Carmel, B. Nicholson,	Journal of Global Information Management	2005
P58	The Impact of Stakeholders' Geographical Distribution on Managing Requirements in a Multi-site Organization	D. E., Damian, D., Zowghi	International Conference on Requirements Engineering	2002
P59	The Development of Knowledge Transfer and Collaboration in a Nearshore Software Development Project	M., Wieandt	Information Systems Workshop on Global Sourcing: Services, Knowledge and Innovation	2007
P60	Effective Communication of Software Development Knowledge Through Community Portals	C., Treude, M.A., Storey	The Joint Meeting of the European Software Engineering Conference and the Symposium on the Foundations of Software Engineering	2011
P61	The Impact of Multi-site Software Governance on Knowledge Management	C., Manteli, B., van den Hooff, A., Tang, H., van Vliet	International Conference on Global Software Engineering	2011

Appendix B. Data Extraction Form

Table 11 - Data items extracted from each study and related research questions

#	Data item	Description	Research Questions
D1	Author(s)	The author(s) of the paper.	
D2	Year	In which year was the study published?	Demographic Attributes
D3	Title	The title of the paper.	
D4	Venue	Publication type of the study: including Journal, Conference, Workshop, etc.	Demographic Attributes
D5	Citation count (Google Scholar)	The citation count of the study in Google Scholar.	Demographic Attributes
D6	Research Method	What type of research method is used in the study? It can be case study, ethnography, etc.	Research Design Attributes
D7	Data Collection Method(s)	What method(s) the researchers use for collecting data.	Research Design Attributes
D8	Data Analysis	It records the approach, which is used for analyzing and interpreting the collected data.	Research Design Attributes
D9	Units of Analysis	It describes the basic element (e.g., a project or an organization) that is being analyzed in the study.	Research Design Attributes
D10	Study Context	The study contexts are categorized in industry and non-industry (e.g. student) cases.	Research Design Attributes
D11	Locations of Sites	It records the number and name of sites that are involved in distributed development.	Contextual Attributes, RQ3
D12	Collaboration Model(s)	The collaboration models used between the partners involved in GSD.	Contextual Attributes, RQ3
D13	Organization Size	The size of organization where the research is carried out or the studied project is selected from.	Contextual Attributes, RQ3
D14	Media and Tool	What media and tool the development teams used for knowledge sharing.	RQ2, Demographic Attributes

D15	Knowledge Sharing Challenge(s)	The challenge(s) of sharing knowledge reported in the study.	RQ1
D16	Knowledge Sharing Practice(s)	What knowledge sharing practice(s) are implemented by the distributed teams	RQ2

pre-Print Copy