Global Software Development and Collaboration: Barriers and Solutions

John Noll, Sarah Beecham, and Ita Richardson
Lero, the Irish Software Engineering Centre
Department of Computer Science and Information Systems
University of Limerick, Limerick, Ireland

{John.Noll,Sarah.Beecham, Ita.Richardson}@lero.ie

ABSTRACT

While organisations recognise the advantages offered by global software development, there are many socio-technical barriers that affect successful collaboration in this inter-cultural environment. In this paper we present a review of the global software development literature where we highlight collaboration problems experienced by a cross-section of organisations in twenty-six studies. We also look at the literature to answer how organisations are over- coming these barriers in practice. We build on our previous study on global software development where we define collaboration as four practices related to agreeing, allocating, and planning goals, objectives, and tasks among distributed teams.

We found that the key barriers to collaboration are geographic, temporal, cultural, and linguistic distance; the primary solutions to overcoming these barriers include site visits, synchronous communication technology, and knowledge sharing infrastructure to capture implicit knowledge and make it explicit.

Categories and subject descriptors

D.2.8 [Software Engineering]: Management—*Productivity, Programming teams, Software process models*

Keywords

Global Software Development, Global Software Engineering, Virtual Teams, Collaboration, Inter-cultural Organizations

1. INTRODUCTION

IN TODAY'S global economy, increasing numbers of software engineers are expected to operate in a globally distributed environment [29]. The tremendous take-up of Global Software Development (GSD) has gone hand-in-hand with technical communication advances such as increased use of e-mail and instant messaging, and inexpensive international telecommunication [52]. In addition, the availability of highly skilled software engineers in low cost locations such as Eastern Europe, Latin America and the Far East [18], coupled with the desire to cut costs and take advantage of the benefits of establishing operations close to emerging markets, have all contributed to more and more organisations selecting this strategy. In some cases, application development and maintenance have been outsourced to remote third party organisations. In others, organisations have set up subsidiaries in low cost economies and "off-shored" part or all of their software development to these locations [8, 63].

The growth of GSD means that many software engineers will have to collaborate over geographic, temporal, cultural and linguistic distance, collectively termed "global distance" [6, 29, 33, 39, 56]. In this environment, geographical distance introduces physical separation between team members and management [6], temporal distance hinders and limits opportunities for direct contact and cooperation [1], and cultural distance negatively impacts on the level of understanding and appreciation of the activities and efforts of remote colleagues and teams [9, 60]. The lack of a common native language (known as "linguistic distance") creates further barriers to communication [8, 29, 39].

Today's practitioners, as well as tomorrow's, are likely to work in a Global Software Development environment at some point in their careers. GSD introduces complexities over and above those experienced in local software development [7, 10, 23, 30]. These complexities, in turn, result in barriers to collaboration among globally distributed developers. It is therefore important for practitioners, and educators, to understand the barriers to collaboration introduced by GSD, and potential solutions for overcoming those barriers.

To contribute to an understanding of the complexities of Global Software Development, this paper presents results of a survey of the GSD literature that addresses two research questions: What barriers prevent software development teams from effectively collaborating in a global environment? And, what solutions address these barriers to collaboration?

While process models such as Capability Maturity Model Integration (CMMI®) [17] and IEC/ISO15504 [34] operate successfully in local environments, they do not explicitly provide for the impact of GSD factors [15], especially in relation to socio-technical complexities [10, 23, 29]. To address this shortcoming, the Global Teaming Model [58, 59] was conceived to provide process guidance to organizations embarking on a GSD effort. The Global Teaming Model takes a project management and operational management view of GSD, and recommends specific practices to address two broad management goals related to GSD: defining global project management, and defining management between locations.

Within the Global Teaming Model, collaboration is one of the specific practices associated with the day-to-day running of GSD, under the second goal ("Define Management Between Locations"). There are four practices associated with collaboration; these are defined as follows:

1. "Identify common goals, objectives and rewards."
2. "Collaboratively establish and maintain the work

product ownership boundaries among interfacing locations within the project or organisation."

3. "Collaboratively establish and maintain interfaces and processes among interfacing locations for the exchange of inputs, outputs, or work products."
4. "Collaboratively develop, communicate and distribute among interfacing teams the commitment lists and work plans that are related to the work product or team interfaces."

However, there are barriers that prevent organisations reaching an agreement on how best to implement these four practices. The goal of this study is to systematically review the GSD literature to uncover potential barriers to collaboration, and to identify specific practices that show *how* organisations can overcome these barriers and thereby realize the advantages offered by GSD.

As such, this paper contributes to the GSD body of knowledge by synthesizing empirical findings that support project management in identifying the best ways to address the problems inherent in working in a distributed environment. For example, we look to the literature in order to establish how trust and communication can be built between locations as a lack of trust and poor communication are shown to be barriers to effective collaboration.

We found twenty-six key primary studies that specifically addressed at least one of our research questions. By combining results from several different studies that are investigating these universal problems we can start to generalise the results beyond the single case study that individual studies often describe. In other words, when findings are supported by many disparate studies we have more confidence in their external validity. These solutions are presented in the context provided by our Global Teaming Model that specifies what is required to achieve effective collaboration in GSD.

The study revealed that the key barriers to GSD collaboration are, not surprisingly, geographic distance, temporal distance, and cultural distance. The primary solutions to overcoming these barriers include in-person site visits, synchronous communication technology that provide as much of the "in person" experience as possible, and knowledge sharing infrastructure to capture implicit knowledge and make it explicit. These are discussed in detail in Section 4.

The remainder of this paper is organized as follows: the next section gives background to, and establishes the context for, this study. Section 3 describes the approach used to conduct this study, and Section 4 presents the results. The paper concludes with a discussion of the results, conclusions, and possible directions for future work.

2. BACKGROUND

This section provides the context for the Global Teaming Model in general, and for the current study of barriers and solutions to collaboration in GSD.

2.1 Virtual Teams

The virtual team is described as the core building block of the virtual organisation [24, 35, 47]. A traditional team is defined as a social group of individuals who are collocated and interdependent in their tasks; the group undertakes and coordinates their activities to achieve common goals and share responsibility for out-comes [55]. Virtual teams have the same goals and objectives as traditional teams and interact through interdependent tasks, but operate across geographic, temporal, and organisational boundaries [43]. They often operate in a multicultural and multilingual environment [25]; communication between virtual team members is normally electronic and often asynchronous, with limited opportunities for informal, face-to-face contact [43]. A virtual team may function on a temporary basis contingent on the demands of the business environment in which it is operating. Regardless, the overall objective is still to function as a single team, with the same goals as if they were collocated.

The implementation of a virtual team strategy can simply be a cost-based decision. This can be achieved by combining the technical skills and experience of staff located in a high-cost centre with engineers in a low-cost location. If the goal is a short-term strategy, then it may be used simply as a knowledge transfer exercise. If, on the other hand, it is a long-term objective, sustained support will be required for team members at all locations. The reason for choosing a particular offshore country can also be based on access to local knowledge or proximity to the customer base [38].

While the term 'distributed team' simply states the geographical location of the team members in the same organisation, the important difference between a virtual team and a distributed team depends on the interdependence of tasks. All virtual teams are considered 'distributed'. However, it is possible to have a team which is geographically distributed, but where the work has been partitioned in such a manner that there is no interdependence of tasks between team members. In these circumstances this team is distributed, but not virtual. As such, the Global Teaming Model is specifically targeted to virtual teams, where there is clear interdependence of tasks between team members at different locations.

2.2 The Global Teaming Model

Effective software project management in a single location is a complex endeavour [4]. There is the need to be an arbitrator between diverse stakeholders with different expectations and agendas, to manage the operation of the team effectively within the constraints of available resources, both financial and technological, and to manage the available personnel and technical capabilities. Therefore, successful software project management is a difficult undertaking which can only be achieved through the effective planning, organizing, staffing, leading, controlling, coordinating and day-to-day management of the project.

Software project management becomes even more complex in a globally distributed environment [36, 53]. In addition to the effective organization and management of collocated teams and projects, there are additional factors which emanate directly from the operation of geographically distributed virtual teams and their related projects. As stated by Paré and Dubé, "The complex, usually uncertain, and highly interdependent nature of project tasks, together with geographical, temporal, structural and cultural gaps fundamental to distributed teams, make management of virtual projects a relatively complex undertaking" [53].

The essence of effective management is coordination and control. But distance introduces complexity which directly influences both coordination and control through its impact on communication and cooperation [6, 50]. Distance also affects project visibility. The need for effective collaboration and visibility between locations in a GSD team environment are essential [36]. These factors are all impacted by language, culture and process and have a direct influence on the level of cooperation which can be achieved. The level of visibility and cooperation within the team must be more controlled. This has to be accomplished under the financial and technical constraints of the project and with team members from geographically dispersed groups who are culturally and linguistically diverse. These global distance issues are not dealt with in structured process models such as CMMI® [17] and IEC/ISO15504 [34].

Previous empirical research involving three independent studies resulted in the definition of twenty-five factors which were shown to influence the effective implementation of GSD [10]. The elicitation and definition of these factors has been reported extensively [9–13, 15, 16, 58].

The Global Teaming Model brings together findings from our empirical research that identified practices that can affect the management of virtual teams. The Global Teaming Model has two specific goals: "Define Global Project Management" and "Define Management Between Locations." Each goal in turn has specific practices and sub-practices; Table 1 lists the goals and practices. We only note issues that relate to Project Management in a global situation, which are over and above the Project Management sub-practices aimed at collocated teams, already present in, for example, the CMMI® that views Project Management in terms of sub-processes like project planning, project monitoring and control, supplier agreement management, etc.

[PLACE TABLE 1 HERE]

While some of the non-technical factors have been recognised previously (e.g. communication, risk management), other social factors, such as fear and trust, were new to research [10, 14], and have now been more widely recognised as important to GSD [46]. Perhaps more critical to GSD is the observation that these factors can have a compounding effect on each other, increasing the possibility of a negative impact on the development process. For example, skills management is further complicated when there are language difficulties across global distance. Knowledge transfer will not occur smoothly unless a team spirit or 'teamness' exists between the individuals. Communication difficulties arise if roles and responsibilities have not been explicitly defined. Therefore, these factors need to be explicitly considered when implementing a GSD strategy.

3. METHOD

For this study, we have taken a systematic, yet focused approach to gathering relevant papers. We do not aim to uncover *all* the recorded problems, but to select a sufficient collection of studies to allow us to identify recurring themes.

Systematic review guidelines [37] recommend the following steps:

1. Identify the need for a systematic literature review.
2. Formulate review research question(s).
3. Carry out a search for relevant studies.
4. Assess and record the quality of included studies.
5. Classify data needed to answer the research question(s).
6. Extract data from each included study.
7. Summarise and synthesise study results (meta-analysis).
8. Interpret results to determine their applicability.
9. Write-up study as a report.

This study conformed largely to these guidelines, with some modifications as discussed below.

Need for a Review.

We have previously undertaken an extensive empirical study that focused on problems encountered in GSD [58]. One of the outputs of this in-depth study was the Global Teaming Model (discussed in Section 2) where we define what is required to collaborate in a virtual environment.

Subsequently, we have examined the software engineering literature and have not found a comprehensive survey that addresses the research questions (stated below) regarding barriers to collaboration, and solutions to overcoming those barriers; hence, the need for this review.

Research Questions.

We seek to answer two research questions:

RQ1: What are the barriers that prevent software development teams from collaborating in a global environment?

RQ2: What solutions are there to addressing the barriers to collaboration, as identified in the literature?

Search.

We used the following Boolean search string to ensure we captured a wide variety of papers related to collaboration in global software development:

```
"Global Software Development" AND
  "Collaboration"
```

We used this search string to search the metadata related to journals and conference proceedings in the *IEEEXplore* bibliographic database.

The search produced thirty-two references. Through reading each paper, we found that six of these did not match our criteria for inclusion (see below). For example, several papers were introductions to workshops, or were not empirical studies. This left us with a final list of twenty-six papers that we have used to answer our research question.

Document selection.

Inclusion and exclusion criteria were used to select the subset of papers from those revealed by the initial search, that should be included in the analysis of the research questions:

We include texts that:

1. directly answer either (or both) of our research questions;
2. were published between January, 2000 and October, 2009;
3. were empirical studies;
4. were full research papers published in a journal or conference proceedings (e.g. not an editorial or introduction to a workshop)

Before accepting a paper into the final set for review, we checked for repeated studies to ensure there is no duplication; for example if the same study is published in two different journals with different first authors, only one study would be included in the review; usually the most comprehensive study or the most recent study.

Kitchenham recommends using a study quality assessment check-list to assess the quality of studies for inclusion in a systematic review, including the quality of the research method.

Since the current study is an attempt to identify themes, rather than establish statistically valid conclusions, the quality criteria for inclusion in the current study are straightforward, so we did not create such a checklist.

Data extraction, meta-analysis, and interpretation.

We examined each selected publication to extract the following elements:

1. Study aims or research question;
2. Identified barriers to collaboration;
3. Identified solutions to the barriers;
4. Other results relevant to the study;
5. Potential themes emerging from the study's conclusions.

Then, we synthesized the data by first identifying major categories of barriers and solutions reported in each selected paper. Subsequently, a summary was created showing the number of papers mentioning each category of barrier and solution (see Table 3 and Table 4). We give each occurrence the same weight, so the frequencies merely reflect how many papers mention a given collaboration barrier or solution, not how important it might be.

4. RESULTS

Of the thirty-two papers examined, twenty-six met the criteria established in Section 3 to serve as sources for this study; these are listed in Table 2. Analysis of these twenty-six papers with respect to our research questions revealed numerous barriers to successful global software development, as well as a variety of solutions to overcome these barriers.

[PLACE TABLE 2 HERE]

4.1 Barriers

The four collaboration practices enumerated in Section 1 depend on accurate communication, motivation toward common vision and goals, cooperation in developing processes and artifacts, and accurate shared understanding of those processes and artifacts.

Barriers to these practices fall into eight categories: geographic distance, temporal distance due to being located in different time zones, differences in language and culture, fear and trust, problems stemming from organizational structure, process issues, barriers deriving from infrastructure, and barriers due to product architecture (see Table 3).

[PLACE TABLE 3 HERE]

Geographic Distance.

Geographic distance introduces numerous barriers to collaboration, the most immediate being the lack of informal encounters that provide not only the opportunity to exchange implicit knowledge, but also to develop personal relationships [3, 5, 7, 19–21, 28, 40,41, 45, 48]

For example, Damian observed that distance impedes "awareness" of remote teams participating on a global project [21]. This affects both the shared understanding about issues like requirements, and the level of trust accorded to remote teams [49].

Distance also increases organizational complexity [7, 20].

Temporal Distance.

Another frequently identified barrier is the time zone difference encountered when development is distributed around the globe. Often termed "temporal distance," the main problem with having developers working in different time zones is that there are fewer hours in the work day when multiple sites can participate in synchronous meetings. Of the twenty-six analyzed, thirteen identified temporal distance as a barrier to collaboration.

One consequence of temporal distance is delay in response to asynchronous communication. This happens when an inquiry sent from one site arrives after working hours at the destination; as a consequence, the response cannot be sent until the next day begins. For example, Nguyen and colleagues observed that the median response to comments posted on a discussion forum was over half a day longer in a distributed context as compared to a single site context [49]. Likewise, Herbsleb and colleagues observed a mean delay in receiving a response to an inquiry in a distributed development context that was over two-and-a-half times as long as that seen in a single-site context [31]; similarly, they observed that "modification requests" (requests for changes to a software component) took over twice as long to be completed in a distributed context [30].

Culture and Language.

English has become the *lingua franca* of global software development [44]. This affects not only the quality of communication, but the choice of communication media. For example, team members who are not confident with their English language skills may prefer instant messaging or email over telephone or video conferencing, as text-based media provide more time to comprehend and compose a response [51]. But text-based media do not convey the visual or auditory queues that convey important information such as how well a participant truly understands a conversation [51].

Language skills can impede communication in more subtle ways. When parties to a conversation have different levels of proficiency, the stronger party occupies a more powerful position and can appear to be more powerful, and thus suppress important communication through unintended intimidation [44, 51]. Further, lack of proficiency in the chosen language can lead to a preference for asynchronous communication, which can be an impediment if video and teleconferencing are important communication media [42].

Culture influences interpretation of communication. For example, polite expressions of acknowledgement by Asian engineers can be misinterpreted as agreement or committment by their European and American colleagues [32, 42]. Culture also affects interpretation of requirements; domain knowledge used to fill in gaps or place requirements in context varies considerably across national culture [32].

Culture also interferes with collaboration when cultural norms result in conflicting approaches to problem solving. For example, American and European engineers were observed to have different views on the value of "up front" design, with Europeans tending toward more initial design effort, and Americans preferring to proceed quickly to implementation; the difference in approach caused conflict and negative impressions [32]. Cultural differences can occur even when teams share a common language and nationality; differences in "corporate culture" can lead to conflicting approaches to problem solving and communication, which in turn might be misinterpreted as rudeness or incompetence [32].

Fear and Trust.

Geographic, temporal, and cultural distance have a significant impact on trust among globally distributed team members. The lack of informal face-to-face meetings means team members have less opportunity to form personal relationships that improve trust among individuals [54]. Delayed response to communication is seen as an barrier to developing ‘familiarity’ among participants, which in turn is seen as an impediment to developing trust [32, 49]. Fear about the future, of jobs and roles, was seen to erode trust as well [32, 42, 54]: on-site teams in expensive countries, fearful of their job security when off-site teams were added in less expensive locations, tended to mistrust their off-site colleagues as well as their own management’s motives [54].

Distance hampers informal communication across sites, making it difficult to disseminate implicit knowledge [5]. This, in turn, limits awareness of remote team’s activities and progress: when it becomes difficult to simply ask about the status of a module or task, there is a greater burden on formal artifacts and process to convey the status of project tasks and deliverables. There is a general tendency to view remote sites as less capable [42]; when formal documentation is missing or perceived to be late, a negative impression of the remote team’s competence may emerge or be reinforced, based on assumptions made by the local team. As Piri and colleagues observed, “... on-site people felt repeatedly uninformed about how the off-site was doing with their tasks. Because of weak trust, this caused general skepticism towards the working morale and commitment of the off-site people” [54]

Organization, Process, and Management

Global software development introduces organizational challenges beyond distance and cultural differences. Distribution development may split the groups with decision-making and project-execution power, and the group with the knowledge of the stakeholder needs [3]. Follow-the-sun development requires greater coordination among teams, because time differences make it difficult to seek clarification on work in progress after it is handed to the next team [7].

Distributed teams place more burden on process than their single-site counterparts. Lack of implicit knowledge resulting from limited informal communication means processes like change management, if not explicitly and thoroughly defined, can be applied differently at different sites [3, 48]. Distributed development also introduces the possibility of different (and incompatible) processes at various sites [3]. Lack of informal communication also limits process visibility, leading to misunderstandings and frustration on the part of remote teams [5, 48].

Infrastructure

Geographic distance means face-to-face meetings are not possible, so distributed teams must rely on video and teleconferencing infrastructure to communicate; even intermittent failures of these technologies can mean that communication simply does not take place [62].

Geographic and temporal distance also mean that implicit knowledge that would normally be disseminated through informal communication must be made more explicit. Inadequate knowledge management infrastructure can inhibit the formation of a shared understanding among project teams [5, 22].

A project with multiple distributed teams introduces the possibility of multiple, incompatible data repositories, which has a risk of data loss when transferring from one repository to another [3].

Product Architecture

The allocation of components to teams affects the productivity of distributed development. Architectures that require multiple sites to implement a change increase the time required to complete the change [31]. An unstable architecture that changes while development work is ongoing can cause confusion among distributed teams as to what their responsibilities are [48]; changing architecture can also cause integration tests to become out-of-sync with the emerging product, resulting in spurious test failures [48].

Product structure can also introduce subtle obstacles among collaborating teams: when a component contains high-visibility functionality, cooperation may be replaced by competition among teams to claim ownership of the component; the inverse would be true if a component is risky or error-prone [32].

4.2 Solutions

The surveyed literature provides numerous solutions to address the barriers identified above, with varying amounts of empirical evidence to support them. These can be classified into seven solution areas, as: approaches to address language and cultural differences; promoting trust and overcoming fear; improving communication infrastructure; management interventions; organizational structures; addressing distributed development process issues; and choosing appropriate product architecture (see Table 4).

[PLACE TABLE 4 HERE]

Culture and Language.

One approach to culture and language differences is to choose sites in culturally similar locations, for example America and Ire land [7, 45]. Absent this, documents authored by non-native speakers of the shared language should be reviewed by a native speaker [42]. Another approach is to develop a shared organizational culture that comprises shared vision and processes, and is developed collaboratively by all participants [3, 45].

The most common approach to dealing with cultural differences is to develop understanding of different cultures through interaction with team members from remote teams, via site visits or face-to-face meetings using video-conferencing technology. Or, cultural awareness can be introduced into the on-site team by including a member from each of the remote teams' cultures, to serve as "cultural ambassadors" who can both interpret remote team communication and actions, and serve as a contact for remote developers, who will perceive the cultural ambassador as someone who understands their language and culture [7, 42, 64]. Further, managers can be rotated to different locations to acquire first-hand knowledge of how different sites work [42].

Infrastructure.

Geographic distance means face-to-face meetings are not possible, so distributed teams must rely on video and teleconferencing infrastructure to communicate; even intermittent failures of these technologies can mean that communication simply does not take place [62].

Geographic and temporal distance also mean that implicit knowledge that would normally be disseminated through informal communication must be made more explicit. Inadequate knowledge management infrastructure can inhibit the formation of a shared understanding among project teams [5, 22].

A project with multiple distributed teams introduces the possibility of multiple, incompatible data repositories, which has a risk of data loss when transferring from one repository to another [3].

Communication technology is the most commonly cited approach to the geographic distance problem. Synchronous communication technology such as video conferencing attempts to replicate the rich interaction present in face-to-face meetings, and is seen as the best solution for meetings in which reaching agreement is the objective [3].

However, some interactions are highly technical in nature, involving intricate content such as program source code In such cases, text-base media such as email, discussion forums, and even log entries in source code control systems, are more appropriate as they can accurately transmit the fine details of such content [42]. The choice of which communication medium is most appropriate for a given situation depends on more than the information to be communicated and the temporal distance between parties. Several studies note that language skills, culture, and personality affect individual preferences for communication technology. A person's cognitive or learning style can influence the choice of communication media; based on their observations of team members in distributed contexts, Aranda and colleagues[2] propose a mechanism for choosing technology based on a model of learning style.

Communication media are commonly classified as synchronous and asynchronous. But in reality, different media fall on a continuum according to the immediacy and richness of interaction. At the same time, various media have different strengths in communicating certain kinds of information. For example, email, wikis, and to some degree instant messaging, are especially useful for conveying detailed technical information such as program source code [42]. Conversely, rich synchronous communication technology such as video-conferencing is appropriate for highly interactive discussions where body language and intonation can convey the degree of understanding or agreement among participants [20, 22].

Language skill can also influence media choice. Programmers with weaker English mastery prefer asynchronous forms of communication (such as email or instant messaging) over telephone or video conferencing, because they provide more time to compose a response; even relatively interactive technology like chat provides enough delay for this purpose [32, 42].

In addition to communication media, a common set of tools shared by all sites, including a configuration management system that stores design documents and meeting minutes, as well as source code, facilitates collaboration [42].

Management.

The most commonly cited management technique involved face-to-face meetings, either virtual (via video conference) or in person, by having remote team members visit the onshore site, and vice-versa. This has a number of benefits. Face-to-face communication provides visual cues as well as highly interactive conversation, which improves understanding [62]. Team members develop personal relationships, which improves trust as well as awareness [22, 32, 42, 54].

Several studies recommend a face-to-face project "kickoff" meeting at the beginning of a project [20, 22, 42, 48]. Also, human facilitators are seen as important to get the most out of video communication technology [3, 22, 42].

Strategic decisions affect the success of a global software development project. It is important to recognise that the barriers outlined above represent risks to project success, and so global software projects should be undertaken only when potential benefits outweigh risks [42, 45]. Stable projects with well-understood objectives and mature teams are the best candidates [42]. Also, the peculiarities of global software development have to be recognised and planned for early in the project life-cycle, as delays reduce the ability to react to adverse events [19, 32, 42].

Organization.

The allocation of tasks and responsibilities to distributed teams can have a significant impact on project success. Because of the communication barriers introduced by distance, coordination among sites is more complicated and difficult than would be the case for co-located teams [45]. Consequently, responsibilities for each site should be chosen to reduce coordination requirements. For example, offshore teams can be given maintenance tasks or other sup- port responsibilities that require less coordination with the main development activity; or, they can be given responsibility for entire products or subsystems so they can act with a high degree of autonomy [7, 42].

Temporal distance can be addressed by choosing locations in the same or adjacent time zones, such as off-shoring from North to South America, or Western Europe to Eastern Europe or Africa [7,42].

There seems to be little research into the best composition of personnel for global software development teams. Aside from the aforementioned cultural ambassadors at the primary site (on-site or controlling site), two studies suggested the need for a technical lead to serve as a representative or deputy of the project manager at each site [19, 62].

Process.

As mentioned above (under “Temporal Distance”), distributed development across time zones can introduce delays. Specific process elements can reduce these effects to some extent by specifying how inquiries should be handled, for example, requiring an immediate response to inquiries received via chat [49]. Also, delay can be mitigated by processes that include frequent iterations and delivery of code increments [3, 19], and escalate problems immediately [19, 45].

Regardless of the specific processes employed, it is important to ensure that all sites follow a shared, agreed process [3, 42], and participants receive training on process elements [3, 20, 22, 48].

Fear and Trust.

Trust can either strengthen or undermine collaboration among global teams [3]. Trust is undermined by fear of the consequences of introducing remote development on the part of the “home” team members, who may feel that their jobs are threatened when work is transferred to lower cost countries; also, trust is affected by perceptions of lack of skill or commitment on the part of remote team members [54].

The most common approach to developing trust among global software developers is to arrange face-to-face meetings, either via video conference or in-person [3, 20, 22, 32, 48, 54]. This allows team members to form personal relationships.

Lack of awareness of what remote teams are achieving, and the difficulties they are experiencing, can have a negative effect on trust [54]. Thus, trust can also be fostered by increasing awareness of what various teams are doing, through wikis and other knowledge sharing infrastructure [5, 22].

Trust can be affected by fear of the future on the part of on-site team members; this can be addressed by clearly expressing the real reasons for introducing off-site development [54].

Product Architecture.

A modular product structure reduces communication overhead by isolating functionality in individual components that can be developed at a single site [29]. A stable architecture is also necessary, to facilitate clear task allocation, reduce replanning, and ensure integration tests match code structure [48].

5. DISCUSSION

As mentioned in Section 2, we have identified four practices comprising collaboration in global software development:

1. Identify common goals, objectives, and rewards.
2. Collaboratively establish product ownership.
3. Collaboratively establish interfaces and processes.
4. Collaboratively develop work plans.

Each facet requires effective communication among sites, to ensure that all participants have a common understanding of the relevant issues.

Each facet also depends on mutual trust among sites and participants, as collaboration is necessarily based on mutual agreement and consensus, rather than authority.

Finally, implicit in the four collaboration facets is the notion that participants will enthusiastically strive toward the agreed goals, develop the agreed products owned, faithfully carry out the established processes, and perform the tasks specified in agreed work plans. Delays, lack of trust, and lack of shared understanding can impede both the actual and perceived performance of these activities. Three themes recur throughout the works surveyed:

1. Geographic and temporal distance limit informal communication, which in turn impedes the building of trust among distributed teams, limits the degree to which implicit knowledge is shared among teams, and interferes with the ability to solve process issues.

2. Cultural differences cause technical misunderstandings, of goals, tasks, requirements, etc., and inhibit the formation of trust due to lack of shared understanding among teams.

3. Temporal distance also results in delays, in completing tasks and responding to inquiries; this can lead to incorrect assumptions and again, mistrust among teams due to perception of lack of commitment.

Distance, Time, and Informal Communication.

Geographic and temporal distance inhibit informal communication: there are few opportunities for chance encounters among team members at different sites, and time zone differences limit opportunities for synchronous communication.

This has several consequences: implicit knowledge that would be disseminated through informal communication is limited, leading to lack of shared understanding of project goals, requirements, and tasks, and lack of trust among distributed teams due to weaker personal relationships among team members. Also, reliance on technology such as instant messaging, email, telephone and video-conferencing for communication precludes non-verbal communication, which can lead to misunderstanding.

Culture, Communication, and Trust.

Differences in language proficiency among distributed team members create barriers to effective communication: different sites may interpret communication in different ways, influenced by their native language and culture; team members with more proficient language skills may lack confidence in their remote counterparts' understanding of communication; less proficient team member may feel inhibited from asking for clarifications due to fear of looking stupid, resulting in incorrect assumptions.

Differences in culture can lead to lack of trust among teams, especially when accompanied by fear of job security resulting from the addition of remote teams in low labor cost locations. Cultural differences also inhibit shared understanding of goals, requirements, and tasks, resulting in rework and delay.

Temporal Distance and Delay.

Delays result from time zone differences and reliance on asynchronous communication technology, for example when seeking clarification of formal documents: an inquiry submitted to an Indian team from the east coast of North America near the end of the work day will likely not be answered until the beginning of the next work day, as the Indian team will not have started work before the Americans go home.

Delays also result from the nature of communication over distance: asynchronous communication media such as email, discussion forums, and wikis have built-in delay, and allow the responder to introduce additional delay.

5.1 Limitations

We depend on papers that may not have presented work that has external validity; although all of the papers included in this review were empirical studies, they tend to depend on one or two case studies to support their findings. However, by combining several studies we start to gain a more general idea of the problem domain.

Our search string was intentionally constructed to produce a highly-focused set of candidate papers for review. By including additional terms in the search string (for example, by adding "global software engineering"), and searching additional libraries, we might have produced a larger initial pool of candidates. *IEEEXplore* comprises a broad array of literature from conferences and journals, ensuring that our findings represent a cross-section of available results. Although our search was focused, it revealed multiple studies discussing each of the barriers and solutions we have identified. Broadening either the search terms or target libraries might reveal some additional barriers or solutions. More likely, however, a larger candidate pool would merely produce additional evidence in support of the barriers and solutions we have already identified.

6. CONCLUSIONS

This study began with a definition of collaboration based on a Global Teaming Model derived from empirical studies of global software development [58]. This model defines collaboration in global software development as having four "practices": identifying common goals, objectives, and rewards; collaboratively establishing product ownership; collaboratively establishing interfaces and processes; and, collaboratively developing work plans.

Two research questions follow from this definition:

> RQ1: What are the barriers that prevent software development teams from collaborating in a global environment?
>
> RQ2: What solutions are there to addressing the barriers to collaboration, as identified in the literature?

Using a systematic approach to reviewing existing empirical studies of global software development, we identified both barriers and solutions that have been observed in practice. These form the answers to our research questions, summarized as follows.

RQ1: What are the barriers that prevent software development teams from collaborating in a global environment?

Eight categories of barriers were identified: geographic distance, temporal distance, linguistic and cultural distance, fear and trust, problems stemming from organizational structure, process issues, barriers deriving from infrastructure, and barriers due to product architecture (see Table 3).

More fundamentally, geographic, temporal, linguistic, and cultural distance interfere with informal communication; fear and lack of trust further inhibit remaining opportunities for communication. This situation, in turn, amplifies issues stemming from organizational structure, process breakdowns, infrastructure, or product architecture, that might otherwise be addressed by a face-to-face meeting with a colleague down the hall.

RQ2: What solutions are there to addressing the barriers to collaboration, as identified in the literature?

Seven categories of solutions emerged: approaches to address language and cultural differences; techniques for promoting trust and overcoming fear; communication infrastructure; management interventions; organizational structures; and distributed development processes.

In effect, solutions attempt to overcome distance issues in three ways: by providing as much of the "in person" communication experience as possible over distance, by adapting processes and organizational structure to address delays stemming from temporal distance, and by providing infrastructure and process to promote shared knowledge and understanding that would naturally develop from informal meetings in a co-located setting.

But one "solution" is to consider the appropriateness of the GSD approach at the outset. As Cusick and Prasad note, "Not all projects are well suited to an offshore model": the complexity of the project, including coordination requirements, process and organizational issues, and risks, must be considered and addressed at the outset, and may outweigh the potential benefits of GSD [19].

6.1 Future Directions

The studies analyzed for this paper examined what might be termed "conventional" global software development: onshore-offshore arrangements or distributed teams that are part of global corporations. Open source software projects represent a different model of global software development that seems to overcome many of the barriers discussed in this paper, with relatively simple tools and processes [65]. Given that many large corporations are now participating in and even sponsoring open source projects, it would be useful to examine the organizational, process, and infrastructure solutions employed by such projects to overcome the barriers of GSD.

7. ACKNOWLEDGMENTS

The research presented in this paper has been supported in part by Science Foundation Ireland through CSET grant no. 03/CE2/I303.1, and by Lero - the Irish Software Engineering Research Centre.

References

[1] P. J. Ågerfalk and B. Fitzgerald. Flexible and distributed software processes: old petunias in new bowls? *Communications of the ACM*, 49(10):26–34, 2006.

[2] G. N. Aranda, A. Vizcaino, A. Cechich, and M. Piattini. Technology selection to improve global collaboration. In *Proceedings of the IEEE International Conference on Global Software Engineering (ICGSE '06)*, pages 223–232, Florianopolis, Brazil, Oct. 2006. IEEE Computer Society.

[3] J. M. Bhat, G. Mayank, and S. N. Murthy. Overcoming requirements engineering challenges: Lessons from offshore outsourcing. *IEEE Software*, 23(5):38–44, 2006.

[4] B. W. Boehm and R. Ross. Theory-W software project management: Principles and examples. *IEEE Transactions on Software Engineering*, 15(7):902–916, 1989.

[5] B. Bruegge, A. H. Dutoit, and T. Wolf. Sysiphus: Enabling informal collaboration in global software development. In *Proceedings of the IEEE international conference on Global Software Engineering (ICGSE '06)*, pages 139–148, Florianopolis, Brazil, Oct. 2006. IEEE Computer Society.

[6] E. Carmel. *Global Software Teams: Collaboration Across Borders and Time Zones*. Prentice Hall PTR, Upper Saddle River, NJ, 1999.[7] E. Carmel and R. Agarwal. Tactical approaches for alleviating distance in global software development. *IEEE Software*, 18 (2):22–29, 2001.

[8] E. Carmel and P. Tjia. *Offshoring Information Tecnhnology: Sourcing and Outsourcing to a Global Workforce*. Cambridge University Press, Cambridge, U.K., 2005.

[9] V. Casey. Leveraging or exploiting cultural difference? In *Proceedings of the 2009 Fourth IEEE International Conference on Global Software Engineering (ICGSE '09)*, pages 8–17, Limerick, Ireland, 2009. IEEE Computer Society.

[10] V. Casey. *Software Testing And Global Industry: Future Paradigms*. Cambridge Scholars Publishing, Newcastle, UK, 2009.

[11] V. Casey and I. Richardson. A practical application of the IDEAL model. *Software Process Improvement and Practice*, 9(3):123–132, 2004.

[12] V. Casey and I. Richardson. Project management within virtual software teams. In *Proceedings of the IEEE International Conference on Global Software Engineering (ICGSE '06)*, Florianopolis, Brazil, Oct. 2006. IEEE Computer Society.

[13] V. Casey and I. Richardson. Uncovering the reality within virtual software teams. In *GSD '06: Proceedings of the 2006 international workshop on Global software development for the practitioner*, pages 66 –72, Shanghai, China, 2006. ACM Press.

[14] V. Casey and I. Richardson. The impact of fear on the operation of virtual teams. In *Proceedings of the 2008 IEEE International Conference on Global Software Engineering (ICGSE '08)*, pages 163–172, Bangalore, India, Aug. 2008. IEEE Computer Society.

[15] V. Casey and I. Richardson. A structured approach to global software development. In R. O'Connor, N. Baddoo, K. Smolander, and R. Messnarz, editors, *Proceedings, Software Process Improvement: 15th European Conference (EuroSPI 2008)*, Dublin, Ireland, 3rd-5th Sept. 2008. Springer Publishing.

[16] V. Casey and I. Richardson. Virtual teams: Understanding the impact of fear. *Special Issue: Global Software Development: Where Are We Headed? Software Process: Improvement and Practice*, 13(6):511–526, 2008.

[17] CMMI Product Team. Capability maturity modelQR integration for development. Technuical report, Software engineering Institute, 2006.

[18] G. B. Crow and B. Muthuswamy. International outsourcing in the information technology industry: Trends and implications. *Communications of the International Information Management Association*, 3(1), 2003.

[19] J. Cusick and A. Prasad. A practical management and engineering approach to offshore collaboration. *IEEE Software*, 23(5):20–29, 2006.

[20] G. A. Dafoulas, K. Swigger, R. Brazile, F. N. Alpaslan, V. L. Cabrera, and F. C. Serce. Global teams: Futuristic models of collaborative work for today's software development industry. In *Proceedings of the 42nd Hawaii International Conference on System Sciences (HICSS '09)*, Big Island, Hawaii, USA, 5-8 Jan. 2009. IEEE Computer Society.

[21] D. Damian, S. Marczak, and I. Kwan. Collaboration patterns and the impact of distance on awareness in requirements-centred social networks. In A. Sutcliffe and P. Jalote, editors, *Proceedings, 15th IEEE International Requirements Engineering Conference (RE '07)*, pages 59–68, New Delhi, India, 15-19 Oct. 2007. IEEE Computer Society.

[22] D. E. Damian and D. Zowghi. The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization. In *Proceedings of the 10th Anniversary IEEE Joint International Conference on Requirements Engineering (RE '02)*, pages 319–328, Essen, Germany, 9-13 Sept. 2002. IEEE Computer Society, ACM Sigsoft, GI (German Computer Society), IFIP Worging Group 2.9, INCOSE, IEEE Computer Society.

[23] D. E. Damian and D. Zowghi. An insight into the interplay between culture, conflict and distance in globally distributed requirements negotiations. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS ’03)*, Big Island, Hawaii, USA, 6-9 Jan. 2003. IEEE Computer Society.

[24] W. H. Davidow and M. S. Malone. *The Virtual Corporation*. Harper Business, New York, 1992.

[25] G. DeSanctis, N. Staudenmayer, and S. S. Wong. Interdependence in virtual organizations. In C. L. Cooper and D. M. Rousseau, editors, *The Virtual Organization*, volume 6 of *Trends in Organizational Behaviour*, pages 81–104. John Wiley & Sons., Chichester, 1999.

[26] J. A. Espinosa and E. Carmel. The effect of time separation on coordination costs in global software teams: a dyad model. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS ’04)*, Big Island, Hawaii, USA, 5-8 Jan. 2004. IEEE Computer Society.

[27] T. Forbath, P. Brooks, and A. Dass. Beyond cost reduction: Using collaboration to increase innovation in global software development projects. In *Proceedings of the 2008 IEEE International Conference on Global Software Engineering (ICGSE ’08)*, pages 205–209, Bangalore, India, Aug. 2008. IEEE Computer Society.

[28] O. Gotel, V. Kulkarni, M. Say, C. Scharff, T. Sunetnanta, S. Touch, and P. Des. Quality-driven competition: Uniting undergraduates, graduates and professionals on global software development projects. In *Proceedings of the 2008 21st IEEE-CS Conference on Software Engineering Education and Training Workshop*, pages 19–21. IEEE Computer Society,14-17 Apr. 2008.

[29] J. D. Herbsleb. Global software engineering: The future of socio-technical coordination. In *Future of Software Engineering (FOSE’07)*, pages 188–198, Minneapolis, MN, USA, 2007. IEEE.

[30] J. D. Herbsleb and A. Mockus. An empirical study of speed and communication in globally distributed software development. *IEEE Transactions on Software Engineering*, 29(6): 481–494, 2003.

[31] J. D. Herbsleb, A. Mockus, T. A. Finholt, and R. E. Grinter. An empirical study of global software development: Distance and speed. In *Proceedings of the 23rd International Conference on Software Engineering, 2001 (ICSE 2001)*, pages 81–90, Toronto, Ontario, Canada, 12-19 May 2001.

[32] J. D. Herbsleb, D. J. Paulish, and M. Bass. Global Software Development at Siemens: Experience from nine projects. In *Proceedings, 27th International Conference on Software Engineering - ICSE 2005*, pages 524–533, St. Louis, Missouri, USA, May 2005. IEEE Computer Society Press.

[33] H. Holmstrom, E. ó Conchúir, P. J. Ågerfalk, and B. Fitzgerald. Global software development challenges: A case study on temporal, geographical and socio-cultural distance. In *International Conference on Global Software Engineering (ICGSE ’06)*, 2006.

[34] ISO. ISO/IEC 15504, information technology – process assessment – part 5: An exemplar process assessment model, ISO/IEC JTC1/SC7. Technical report, International Standards Organization, 2006.

[35] S. L. Jarvenpaa and B. Ives. The global network organization of the future: Information management opportunities and challenges. *Journal of Management Science and Information Systems*, 10(4):25–57, 1994.

[36] D. W. Karolak. *Global Software Development: Managing Virtual Teams and Environments*. IEEE Computer Society Press, Los Alamitos, CA, USA, 1999.

[37] B. Kitchenham. Procedures for performing systematic reviews. Technical Report Keele University TR/SE-0401 and NICTA 0400011T.1, Software Engineering Group, Department of Computer Science, Keele University, and Empirical Software Engineering, National ICT Australia Ltd., July 2004.

[38] J. Kotlarsky and I. Oshri. Country attractiveness for off-shoring and offshore outsourcing: additional considerations. *Journal of Information Technology*, 23:228–231, 2008.

[39] S. Krishna, S. Sahay, and G. Walsham. Managing cross-cultural issues in global software outsourcing. *Communications of the ACM*, 47(4):62–66, 2004.

[40] M. T. Lane and P. J. Ågerfalk. On the suitability of particular software development roles to global software development. In *Proceedings of the 2008 IEEE International Conference on Global Software Engineering (ICGSE ’08)*, pages 3–12, Bangalore, India, Aug. 2008. IEEE Computer Society.

[41] S. Lasser and M. Heiss. Collaboration maturity and the offshoring cost barrier: The tradeoff between flexibility in team composition and cross-site communication effort in geographically distributed development projects. In *Proceedings of the International Professional Communication Conference (IPCC 2005)*, pages 718–728, 10-13 July 2005.

[42] B. Lings, B. Lundell, P. J. Ågerfalk, and B. Fitzgerald. A reference model for successful distributed development of software systems. In *Proceedings of the International Conference on Global Software Engineering (ICGSE ’07)*, pages 130–139, Munich, Germany, Aug. 2007. IEEE Computer Society Technical Committee on Software Engineering, IEEE Computer Society.

[43] J. Lipnack and J. Stamps. *Virtual Teams: Reaching Across Space, Time And Organizations with Technology*. John Wiley & Sons, 1997.

[44] B. Lutz. Linguistic challenges in global software development: Lessons learned in an international SW development division. In *Proceedings of the 2009 Fourth IEEE International Conference on Global Software Engineering (ICGSE ’09)*, pages 249–253, Limerick, Ireland, 2009. IEEE Computer Society.

[45] E. MacGregor, Y. Hsieh, and P. Kruchten. The impact of inter-cultural factors on global software development. In *Canadian Conference on Electrical and Computer*

Engineering, pages 920–926, 2005.

[46] N. B. Moe and D. Smite. Understanding a lack of trust in global software teams: A multiple-case study. *Software Process: Improvement and Practice*, 13(3):217–231, June 2008.

[47] S. A. Mohrman. The context for geographically dispersed teams and networks. In C. L. Cooper and D. M. Rousseau, editors, *The Virtual Organization*, volume 6 of *Trends in Organizational Behaviour*, pages 63–80. John Wiley & Sons., Chichester, 1999.

[48] N. Mullick, M. Bass, Z. E. Houda, D. J. Paulish, M. Cataldo, J. D. Herbsleb, L. Bass, and R. Sangwan. Siemens global studio project: Experiences adopting an integrated GSD infrastructure. In *Proceedings of the IEEE international conference on Global Software Engineering (ICGSE '06)*, pages 203–212, Florianopolis, Brazil, Oct. 2006. IEEE Computer Society.

[49] T. Nguyen, T. Wolf, and D. Damian. Global software development and delay: Does distance still matter? In *Proceeings of the 2008 IEEE International Conference on Global Software Engineering (ICGSE '08)*, pages 45–54, Bangalore, India, Aug. 2008. IEEE Computer Society.

[50] K. E. Nidiffer and D. Dolan. Evolving distributed project management. *IEEE Software*, 22(5):63–72, 2005.

[51] T. Niinimäki, A. Piri, and C. Lassenius. Factors affecting audio and text-based communication media choice in global software development projects. In *Proceedings of the 2009 Fourth IEEE International Conference on Global Software Engineering (ICGSE '09)*, pages 153–162, Limerick, Ireland, 2009. IEEE Computer Society.

[52] J. A. O'Brien. *Management Information Systems - Managing Information Technology in the Business Enterprise*. McGraw Hill/Irwin, 6 edition, 2004.

[53] G. Paré and L. Dubé. Virtual teams: an exploratory study of key challenges and strategies. In P. De and J. I. De Gross, editors, *Proceedings of the 20th International Conference on Information Systems*, pages 479–483, Charlotte, North Carolina, USA, Dec. 1999. Association for Information Systems.

[54] A. Piri, T. Niinimäki, and C. Lassenius. Descriptive analysis of fear and distrust in early phases of GSD projects. In *Proceedings of the 2009 Fourth IEEE International Conference on Global Software Engineering (ICGSE '09)*, pages 105–114, Limerick, Ireland, 2009. IEEE Computer Society.

[55] A. Powell, G. Piccoli, and B. Ives. Virtual teams: A review of current literature and direction for future research. *The DATA BASE for Advances in Information Systems*, 35(1):6–36, 2004.

[56] R. Prikladnicki, J. L. N. Audy, and R. Evaristo. Global software development in practice lessons learned. *Software Process: Improvement and Practice*, 8(4):267–281, 2004.

[57] I. Richardson, S. Moore, A. Malone, V. Casey, and D. Zage. Globalizing software development in the local classroom. In *20th Conference on Software Engineering Education and Training (CSEET '07)*, pages 64–71, 2007.

[58] I. Richardson, V. Casey, J. Burton, and F. McCaffery. Global software engineering: A software process approach. In I. Mistrik, J. Grundy, A. van der Hoek, and J. Whitehead, editors, *Collaborative Software Engineering*, pages 35–56. Springer- Verlag/Computer Science Editorial, 2010.

[59] I. Richardson, V. Casey, S. Beecham, J. Burton, and F. McCaffery. Managing virtual software development teams: A process framework. *IEEE Transactions on Engineering Management, special issue on the impacts of global information systems sourcing on engineering, technology and innovation management*, in review.

[60] A. F. Rutkowski, D. R. Vogel, M. van Genuchten, T. M. A. Bemelmans, and M. Favier. E-collaboration: The reality of virtuality. *IEEE Transactions on Professional Communication*, 45(4):219–230, Dec. 2002.

[61] F. C. Serce, F.-N. Alpaslan, K. Swigger, R. Brazile, G. Dafoulas, V. Lopez, and R. Schumacker. Exploring collaboration patterns among global software development teams. In *Proceedings of the 2009 Fourth IEEE International Conference on Global Software Engineering (ICGSE '09)*, pages 61–70, Limerick, Ireland, 2009. IEEE Computer Society.

[62] A. Taweel, B. Delaney, T. N. Arvanitis, and L. Zhao. Communication, knowledge and co-ordination management in globally distributed software development: Informed by a scientific software engineering case study. In *Proceedings of the 2009 Fourth IEEE International Conference on Global Software Engineering (ICGSE '09)*, pages 370–375, Limerick, Ireland, 2009. IEEE Computer Society.

[63] S. S. Toaff. Don't play with "mouths of fire" and other lessons of global software development. *Cutter IT Journal*, 15(11): 23–28, 2002.

[64] R. Urdangarin, P. Fernandes, A. Avritzer, and D. Paulish. Experiences with agile practices in the global studio project. In *Proceedings of the 2008 IEEE International Conference on Global Software Engineering (ICGSE '08)*, pages 77–86, Bangalore, India, Aug. 2008. IEEE Computer Society.

[65] Y. Yamauchi, M. Yokozawa, T. Shinohara, and T. Ishida. Collaboration with lean media: How open-source software succeeds. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, pages 329–338, Philadelphia, Pennsylvania, United States, 2000. ACM Press.

Goal:	**Define Global Project Management**	**Define Management between Locations**
Practice	Global Task Management	Operating Procedures
Sub-practice	Determine team and organisational structure between locations Determine the approach to task allocation between locations	Define how conflicts and differences of opinion between locations are addressed and resolved Implement a communication strategy for the team Establish communication interface points between the team members Implement strategy for conducting meetings between locations
Practice	Knowledge and Skills	Collaboration Between Locations
Sub-practice	Identify business competencies required by global team members in each location Identify the cultural requirements of each local sub-team Identify communication skills for GSD	Identify common goals, objectives and rewards Collaboratively establish and maintain work product ownership boundaries Collaboratively establish and maintain interfaces and processes Collaboratively develop, communicate and distribute work plans
Practice	Global Project Management	
Sub-practice	Identify GSD project management tasks Assign tasks to appropriate team members Ensure awareness of cultural profiles by project managers Establish cooperation and coordination procedures between locations Establish reporting procedures between locations Establish a risk management strategy	

Table 1: Global Teaming Goals and Practices

Author(s)	Title	Ref
Aranda et al.	Technology Selection to Improve Global Collaboration	[2]
Bhat et al.	Overcoming Requirements Engineering Challenges: Lessons from Offshore Outsourcing	[3]
Bruegge et al	Sysiphus: Enabling informal collaboration in global software development	[5]
Carmel and Agarwal	Tactical Approaches for Alleviating Distance in Global Software Development	[7]
Cusick and Prasad	A Practical Management and Engineering Approach to Offshore Collaboration	[19]
Dafoulas et al	Global Teams: Futuristic Models of Collaborative Work for Today's Software Development Industry	[20]
Damian et al	Collaboration Patterns and the Impact of Distance on Awareness in Requirements-Centred Social Networks	[21]
Damian and Zowghi	The Impact of Stakeholders' Geographical Distribution on Managing Requirements in a multi-site ortanization	[22]
Espinosa and Carmel	The Effect of Time Separation on Coordination Costs in Global Software Teams: A Dyad Model	[26]
Forbath et al	Beyond Cost Reduction: Using Collaboration to Increase Innovation in Global Software Development Projects	[27]
Gotel et al	Quality-Driven Competition: Uniting Undergraduates, Graduates and Professionals on Global Software Development Projects	[28]
Herbsleb and Mockus	An Empirical Study of Speed and Communication in Globally Distributed Software Development	[30]
Herbsleb et al	An Empirical Study of Global Software Development: Distance and Speed	[31]
Lane and Ågerfalk	On the Suitability of Particular Software Development Roles to Global Software Development	[40]
Lasser and Heiss	Collaboration Maturity and the Offshoring Cost Barrier: The Tradeoff between Flexibility in Team Composition and Cross-Site Communication Effort in Geographically Distributed Development Projects	[41]
Lings et al	A reference model for successful Distributed Development of Software Systems	[42]
Lutz	Linguistic Challenges in Global Software Development: Lessons Learned in an International SW Development Division	[44]
MacGregor et al	The Impact of Intercultural Factors on Global Software Development	[45]
Mullick et al	Siemens Global Studio Project: Experiences Adopting an Integrated GSD Infrastructure	[48]
Nguyen et al	Global Software Development and Delay: Does Distance Still Matter?	[49]
Piri et al	Descriptive Analysis of Fear and Distrust in Early Phases of GSD Projects	[54]
Richardson et al	Globalizing Software Development in the Local Classroom	[57]
Serce et al	Exploring Collaboration Patterns among Global Software Development Teams	[61]
Taweel et al	Communication Knowledge and Co-ordination Management in Globally Distributed Software Development: Informed by a scientific Software Engineering Case Study	[62]
Urdangarin et al	Experiences with Agile Practices in the Global Studies Project	[64]

Table 2: Papers Accepted for Analysis

Barrier	No. Papers	Citations
Language and Cultural Distance	14	[2, 3, 19–22, 32, 40, 41, 42, 44, 45, 48, 61]
Temporal Distance	13	[2, 3, 5, 20–22, 26, 28, 30, 40, 45, 49, 57]
Geographic Distance	12	[2, 3, 5, 7, 19–21, 28, 40, 41, 45, 48]
Process and Management Issues	11	[3, 5, 7, 19, 21, 28, 31, 40, 54, 48, 62]
Fear and Trust	9	[3, 19, 22, 32, 40, 41, 42, 49, 54]
Infrastructure	7	[3, 19, 22, 40, 45, 48, 62]
Organization	6	[7, 19, 28, 48, 54, 61]
Product Architecture	3	[31, 32, 48]

Table 3: Identified Barriers to Collaboration

Solution Area	No. Papers	Citations
Infrastructure	13	[2, 3, 5, 19–22, 28, 42, 48, 49, 62, 64]
Management	12	[3, 7, 19, 21, 22, 26, 28, 32, 42, 45, 48, 49, 62]
Organization	11	[3, 19, 20, 22, 27, 28, 42, 45, 48, 54, 57, 62]
Process	9	[3, 19, 20, 22, 42, 45, 48, 49, 62]
Fear and Trust	7	[3, 5, 20, 22, 32, 48, 54]
Language and Cultural Distance	6	[3, 7, 19, 20, 45, 64]
Product Architecture	1	[32]

Table 4: Identified Solutions to GSD Collaboration Barriers