International Journal of Computer Science and Applications,
2008, Vol. 5, No. 3b, pp 1 - 24

DISTRIBUTED INFORMATION SYSTEMS DEVELOPMENT: A FRAMEWORK FOR UNDERSTANDING AND MANAGING

JOLITA RALYTÉ[1], XAVIER LAMIELLE[2], NICOLAS ARNI-BLOCH[1], MICHEL LÉONARD[1]

[1]*University of Geneva, CUI, Battelle - bâtiment A, 7, route de Drize,*
CH-1227 Carouge, Switzerland
{Jolita.Ralyte; Nicolas.Arni-Bloch; Michel.Leonard}@ cui.unige.ch
http://cui.unige.ch/~ralyte

[2]*Financial Software Group, Geneva, Switzerland*
xavier.lamielle@bluewin.ch
www.fsggroup.com

Increasing trend towards globalisation and outsourcing of software and information systems (IS) development engenders a number of challenges and difficulties that companies have to deal with. A number of studies have been conducted in order to identify problems as well as potential benefits of distributed systems development. In this paper we propose a framework allowing to characterise the domain of distributed information systems development (DISD) and to help the project manager in assessing DISD project situation, identifying potential problems and evaluating project risks related to different distribution facets. Besides, the framework aims to indicate prospective solutions to each type of problem. It was elaborated on the basis of literature survey and our personal observations in real projects.

Keywords: Distributed IS development, global software development, risk management.

1. Introduction

Distributed development of software and information systems (also named Global Software Development) becomes increasingly common, driven by the globalisation of companies and their business and enabled by new information and communication technologies. Distributed Information Systems Development (DISD) promotes realisation of IS in a collaborative way where several partners, generally situated in distant places, participate in the elaboration of a common solution. The DISD consists in decomposing the IS development process into more or les autonomous phases to be realised by these partners.

DISD aims at increasing enterprise productivity, reducing IS development cost, and enlarging the number of human competencies and skills, which allows not only to share experiences in different cultural environments but also to extend enterprise strategy to the global market. In fact, this kind of practice allows enterprises to deal with new economic globalisation constraints that they have to undergo but it is not devoid of problems. It is clear that the distribution of processes has an impact on the way the IS products will be specified, designed, coded and delivered to the clients.

A number of studies (Boland and Fitzgerald, 2004; Bass and Paulish, 2004; Paasivaara, 2003; Prikladnicki and Yamaguti, 2004) have been conducted by researchers and practitioners and have identified a set of problematic areas common across projects. We can mention spatial and temporal distances, language and cultural difference, lack of trust and shared knowledge. In this work we attempt to position all these problems and difficulties as well as prospective solutions with the help of a framework. The framework aims to help the DISD project manager in analysing project situation, identifying potential problems due to the distributed project environment and therefore evaluating project risks. The framework also allows to position prospective solutions to the identified problems even though our survey of literature shows that this domain still needs significant investment of researchers and practitioners.

The DISD is not anymore limited to the big multinational organisations but is also adopted by small and medium enterprises (SME). This work was realised in collaboration with a SME providing IT solutions in the domain of finances developed in a distributed environment including partners located in India, England and Switzerland.

The paper is organised as follows: in section 2 we define our framework which is based on the state of the art of the domain and our personal experience. Section 3 elaborates the framework while section 4 illustrates its application in a case study.

2. Defining the Framework

Several authors (Agerfalk et al., 2005; Kruchten, 2004; Wiredu, 2006) already attempted to characterise distributed software development by identifying characteristics common to distributed projects and how these characteristics affect different project management activities. In this section we introduce our framework which extends the achievements of other authors (Agerfalk et al., 2005; Wiredu, 2006) (based on our experience in enterprise) with new characteristics and activities. The aim of the framework is to provide a basis for analysing DISD project situation, identifying potential risks, and hopefully finding solutions. The framework is based on six dimensions and five types of activities that we present in the following.

2.1. *Framework Dimensions*

First of all, the realisation of a distributed project means that different actors are not collocated but geographically dispersed. *Geographical distance* was identified by numerous authors (Agerfalk et al., 2005; Lindqvist et al., 2006; Paasivaara, 2003; Wiredu, 2005) as one of the principal dimensions characterising distributed development. However this dimension is not sufficient to characterise the situation of distribution. The time necessary to reach a distant partner and/or to exchange information as well as the difference of national and/or organisational culture are also considered in the literature as important characteristics that can introduce additional complexity to the DISD project and pose new problems. These two characteristics are defined as *temporal* and *socio-cultural distances* (Agerfalk et al., 2005).

Although these three dimensions allow to identify several problems and introduce barriers and complexity (Casey and Richardson, 2006) that DISD has to undergo, in our opinion, they do not constitute a sufficiently precise classification for the identification of all the problems related to the distribution. In order to characterise DISD we ask the following question: "What are the concepts for which the distribution adds some distance?" Of course, the distance here is considered in its global meaning and not only between two locations. It can also concern the time, things and humans (Bernard, 2006) for example, a period of time between two events, a difference between two entities or a gap of knowledge between two human actors. Based on the survey of literature and our personal observation, we identify three additional dimensions named *technological*, *organisational* and *knowledge* distance. The first one deals with difference of the technology (software and hardware) used by different DISD partners, the second identifies divergence in partners organisational structure and business processes while the third considers human skills and knowledge gap. In the following we discuss and illustrate the six dimensions of our framework.

2.1.1. *Geographical Distance*

Geographical distance represents a spatial distance between two entities (organisations, persons, cities, etc.) measured by the cost and time necessary to cover it. It is also measured in ease of delocalising and relocating and has different facets like the type of transportation, the time necessary for travelling from one location to another, the need to have a visa or other permission. In other words, the geographical distance is measured by the effort that an actor has to provide in order to go to the location of the other actor (Agerfalk et al., 2005).

Even though the geographical distance seems to be symmetric, at least from the geographical point of view (the distance between Geneva and Delhi is the same as between Delhi and Geneva), it is not always the case in the DISD. The effort necessary to cover a geographical distance between two locations can be bigger in one direction than in the other. For example, in our experience, often, it was easier for a Swiss project manager to go to India than to bring a developer form India to Switzerland. Moreover, the distance between two locations depends on the transportation facilities. For example, if the offer of regular flights between two locations is significant, the distance will be considered as lower than in the situation where no offer is at all. The geographical distance seems to be the main source of problems in DISD.

2.1.2. *Temporal Distance*

Agerfalk et al. (2005) define the temporal distance as a "dislocation in time, experienced by two actors wishing to interact". This dimension includes characteristics like hours of work, execution time, response time, etc. The most evident characteristic is the time difference between the time zones of two locations. This difference requires managing work schedules of different entities in order to increase as much as possible their overlap and therefore to facilitate the communication between distant actors. For example, the

temporal distance between Indian and Switzerland is bigger then between England and Switzerland but can be considerably reduced if the corresponding partners accept to adapt their hours of work. Another characteristic related to the temporal distance is the time necessary for executing a task. For example, it takes more time to copy files on a distant server than on a local one. The time necessary to get the response to a query from a distant partner is also important and can slow down the productivity when the response is necessary in order to continue the work.

2.1.3. *Socio-cultural Distance*

The success of a DISD project also depends on the socio-cultural factors that are based on the fundamental difference of the system of values that governs our lives. This distance is very complex and concerns different cultural aspects like organisational, business, language and political culture as well as personal motivations and work ethics. The culture defines the norms, values, expression manners, communication styles and has an impact of the respect of the organisation hierarchy (Kruchten, 2004). Agerfalk et al. (2005) define the socio-cultural distance as a measure of understanding and accepting the norms and values of an actor by another one. The socio-cultural proximity between two actors is not always reciprocal; one actor can more easily accept the ways and customs of the other actor.

2.1.4. *Organisational Distance*

We define organisational distance as a dissimilarity in the organisational structure and business processes of two organisations. This difference can be observed in their organisation of work, management of human resources and their knowledge, and the IS development processes (Bass and Paulish, 2004). It is a potential source of problems in DISD. The difference in partners' enterprise structure, organisation and business processes can pose problems in project coordination. For example, we have identified that an actor can have a difficulty to identify the right person to send some particular information, documents, and files and to ask questions. It is not always evident to ensure the traseability of these exchanges between actors and to raise this information to the right level of the organisational hierarchy. Therefore, the organisational distance is measured by the quality of interactions between different actors and the manner to develop IS products: specify, design, develop, test and deliver.

2.1.5. *Technological Distance*

The large variety of technologies (software and hardware) available for IS development leads to a divergence in their application by different DISD partners. We define technological distance as a difference of applied technologies by two DISD entities. This distance is measured by technical obstacles also named technological accidents (Wiredu, 2005) or technological factors (Bass and Paulish, 2004) that have to be got over in order to use the systems required in the DISD project. This dimension concerns not only the technical infrastructure used for IS development but also the information and

communication technologies. On the one hand, the problems can be raised by the difference of used programming languages, software and hardware platforms and the manner to use them; on the other hand, it concerns data flows and information and data management.

2.1.6. *Knowledge Distance*

Variability and complementarities of human knowledge and skills necessary for IS development are considered as one of the advantages that DISD benefits. However, it can also cause many problems. We define the knowledge distance as divergence of knowledge that two entities have in some particular application domain and for the realisation of some particular task, while this knowledge is necessary for the realisation of their work, communication and coordination of their activities. The divergence can be linguistic (knowledge of the communication language, e.g. English), technical (programming language, development tools, etc.), or professional (knowledge about the business domain for which the IS has to be developed, e.g. financial domain). This distance is completely independent of the geographical distance. For example, a developer can be incompetent to realise a task imposed by a colleague located in the same room if he or she does not have enough knowledge for that. However, this distance is bigger when the two persons are situated in distant locations. The location proximity can help to resolve this problem of lack of knowledge. In fact, the training facilities are immediately accessible for the collocated developer as he or she is already well integrated in the enterprise.

2.2. *Framework Activities*

The impact of distribution on different DISD lifecycle activities was highlighted by several authors (Carmel and Agarwal, 2001; Evaristo et al., 2004; Malone and Crowston, 1994; Mcchesney and Gallagher, 2004). Most of them focus their studies on communication, coordination and control activities within distributed development. The framework proposed by Agerfalk et al. (2005) is also founded on these three types of activities. In our opinion, these activities mainly deal with DISD project management while the engineering activities of the IS product are not considered explicitly in this work. Based on our experience in the DISD projects we argue that IS development and maintenance activities also depend of the distribution characteristics and have to be included in the framework. In the following we present the five types of activity.

2.2.1. *Communication Activities*

Communication is an exchange of complete and unambiguous information; it is necessary for knowledge sharing between different actors with the aim to reach a common understanding (Carmel and Agarwal, 2001). According to Cherry and Robillard (2004), communication "plays a critical role in a software engineering process in the gathering and crystallization of all relevant information in quality software which fulfills the user needs on time and within budget". Communication difficulties appear because of

the fact that distribution transforms the traditional face-to-face communication into a more restrictive and constrained form of it (Agerfalk et al., 2005) generally using new communication tools like video conference, e-mail, messengers, etc. Several researchers have insisted on the non-negligible importance of informal communications (Cherry and Robillard, 2004; Robillard and Robillard, 2000), while others has specifically highlighted the fact that distance in distributed software development is a challenge to informal communications which can generate problems of coordination (Cherry and Robillard, 2004; Grinter et al., 1999; Herbsleb and Grinter, 1999; Herbsleb and Mockus, 2003; Paasivaara, 2003). Scott et al. (2005) aim to resolve communication problems by providing a set of communication patterns.

2.2.2. *Coordination Activities*

Malone and Crowston (1990, 1994) have developed a theory of coordination based on a survey of coordination processes in several disciplines such as organisation theory, computer science, economics, linguistics and psychology. They define coordination as "managing dependencies between activities" and propose a framework which characterises different kinds of dependency between activities and identifies coordination mechanisms that can manage these activities. In software development, coordination is defined as "the act of integrating each task with each organisational unit, so the unit contributes to the overall objective" (Carmel and Agarwal, 2001). Coordination concerns interaction between different actors: the coordination process defines how these actors become interdependent on each other when realising a common objective. Besides, Wiredu (2006) argues that the greatest organising challenge in distributed development is coordination of the interactions between distributed people, processes, information and technology and proposes a framework as a theoretical foundation for research on the coordination in distributed development.

2.2.3. *Control Activities*

Control activities are related to the DISD project management. They concern organisation and management of resources (human, material, budget, time) as well as planning and direction of the IS development with the objective to complete the project in time, on budget and of required quality. Control in distributed environment is even more challenging and requires specific project management and reporting mechanisms. Project management of distributed teams is different of that of local teams. To deal with such a difficulty Casey and Richardson (2006) propose to adopt "virtual team strategy" where team members use different networks (Internet, Intranets, Extranets) to communicate, coordinate and collaborate with each other on tasks and projects.

2.2.4. *Development Activities*

Development of an IS in distributed environment can also be affected by different distribution dimensions. According to Prikladnicki et Yamaguti (2004), DISD "is causing a great impact not only in the market, but also in the way the software products are

conceived, designed, constructed, tested, and delivered to customers". For example, problems can appear when integrating modules realised by distant teams using different (versions of) platforms for their realisation. Here we include all ISD lifecycle activities but we do not analyse each activity separately.

2.2.4. *Maintenance Activities*

Maintenance of an IS represents a set of activities that aim to prevent and/or to correct the IS in order to maintain or to restore its conformance to the requirements specification. We consider maintenance as a post-project activity. However, based on our personal observation, we claim that it is important to plan maintenance activities in case of DISD. In fact, maintenance is a very important step in DISD (Yan, 2005). Because of the geographical distance, maintenance can be difficult while project teams are distributed, for example, if correcting an error on client's site requires some particular skill which is only available at the distant partner's location.

2.3. *The Framework*

Based on the dimensions and activities presented in this section, we define our framework in the form of a matrix where each cell represents the impact of one dimension on one activity. In fact, the aim of the framework is to help the DISD project manager to answer the question: How different dimensions of the distributed development govern the DISD activities? By answering this question the project manager would be able to better assess DISD project situation and evaluate the risk related to the distribution. For that, the framework has to identify what kind of problems could occur in each activity because of a particular dimension? We elaborate each cell of the framework in the next section.

3. Positioning Problems and Potential Solutions in the Framework

Literature on distributed software development identifies numerous problems, also named accidents (Wiredu, 2005), related to the distribution of the development process and only very few research works propose solutions to resolve some of them. With our framework we aim to classify these problems and identify potential solutions. While the framework proposed by Agerfalk et al. (2005) emphasizes on the opportunities offered by distributed development, in our work we focus our attention on the understanding of potential risks in DISD projects and finding prospective solutions. The elaboration of the framework presented below is based on the literature survey and our personal experience in several DISD projects. Of course, the collection of values positioned in the framework is not an exhaustive one. We rather recommend that each project manager complete it with his/her personal experience in DISD projects. Besides, the complexity and the interactions between different dimensions and activities make our choice arguable. Some of the problems are related to several dimensions and/or several activities.

The framework is split in two tables: Table 1 summarises problems and solutions related to the geographical, temporal and socio-cultural distances while Table 2 presents those related to the organisational, technological and knowledge distances.

Table 1. A Framework for assessing DISD (part 1).

Activity	Geographical Distance	Temporal Distance	Socio-cultural Distance
Communication	! Lac of informal communication ! Difficulty to initiate contacts ! Additional cost for travelling !Effort for travelling	! Delayed communication ! Information not transmitted ! Increase of asynchronous communication ! Delayed feedback	! Barrier for synchronous communication ! Different interpretation ! Lack of synchronisation meetings
Coordination	! Reduced trust ! Lack of team spirit ! Lack of awareness * Minutes of meetings * System to share files	! Reduced overlap of working hours ! Lack of synchronisation meetings ! Difficulty to coordinate technical infrastructure	! Delayed decisions ! Perception of hierarchy ! Different enterprise culture ! Different values ! Incompatible views ! Doubt about skills * Proxy persons
Control	! Bad visibility of progress ! Lack of managers' involvement !Planning of meetings ! Change management	! Planning of tasks and resources	! Distribution of tasks ! Interpretation and respect of tasks ! Authority perception ! Difficulty in motivating teams ! Interpretation of project priorities ! Different social and political norms ! Knowledge sharing * Define project priorities and rules
Development	! Product interdependency * Product decomposition	! Loss of time * Adjustment of working hours	! Different analysis and design approaches
Maintenance	! Availability of the right person ! Difficulty to maintain high level support * Agreements with customer * Technical liaison engineer	! Difficulty to have a prompt intervention * Efficient communication with customer * Technical liaison engineer	! Reaction of the customer to the delay * Establish good relationships with the customer

The "!" symbol indicates problems while the "*" symbol marks solutions. All values are explained below in the following sub-sections. We can see that problems in these tables are much more numerous than solutions. It is because of the novelty of the situation and therefore the novelty of the research domain. Our literature survey demonstrates that problems related to the distributed development are still analysed and formalised and only very few solutions are proposed, most of them based on developers' experience, interviews and case studies. Besides, problems related to project realisation and control activities received less attention then communication and coordination activities. Due to the very complex nature of the DISD, our choice of positioning problems and solutions can be questionable. Nevertheless, the framework allows us to have a holistic view of the domain.

Table 2. A Framework for assessing DISD (part 2).

Activity	Organisational Distance	Technological Distance	Knowledge Distance
Communication	! Non respect of information flows * Pre-established communication channels and project priorities	! Different communication tools ! Bad communication quality * Common e-mail server	! Lack of language knowledge ! Lack of specific knowledge
Coordination	! Lack of organisational visibility !Interdependency of modules !Interdependency of development processes ! Conflicts of management * Incremental integration	! Lack of technology standardisation	! Lack of common understanding ! Lack of specific knowledge
Control	! Different working processes ! Different communication channels ! Management of distant resources * Common method	! Homogeneity of resources	! Lack of visibility of skills
Development	! Synchronisation of the development process ! Product decomposition ! Redistribution of resources ! Different development methods ! Requirements changes * Iterative and incremental development * Test-driven development	! Difference of working platforms ! Difference of databases ! Limit design choices * Common test infrastructure	! Lack of domain-specific knowledge ! Lack of formalism knowledge * Trainings
Maintenance	! Lack of standardised documentation	! Different technology for maintenance * Compatible maintenance environnement	! Lack of knowledge about the final product

3.1. *Communication in DISD*

3.1.1. *Geographical Distance*

Communication activities have to face several problems related to the geographical distance. One of them is *lack of informal communication* (Cherry and Robillard, 2004; Lindqvist et al., 2006) between distributed teams. Informal contact is necessary to obtain a good team cohesion and team spirit which are necessary to ensure high-quality collaboration and information exchange (Herbsleb and Mockus, 2003). It was observed that a number of problems were identified, clarified and even resolved during the coffee brakes of the developers situated on the same location.

Communication between distant locations is possible thanks to different communication tools but it is also restricted by the application of these tools and generates the *difficulty to initiate the contact* (Herbsleb and Grinter, 1999). For example, the developer can omit to inform the proper authority about a problem or modification of the system that he/she considers as minor because of this effort required to initiate the contact. Indeed, several interviewed developers pointed out that it is easier to ask a

question to a colleague sitting nearby than to contact a person situated in a distant location.

To ensure good communication, a minimum of face-to-face meetings is indispensable. Travelling generates *additional cost* and consumes some time (Battin et al., 2001) which depends on the existing transportation offer. Besides, the *effort necessary for travelling* can pose a problem if the concerned persons are not motivated enough. There is a tendency to minimise the problem if it is necessary to travel for resolving it even though the delay in the project is generally more costly than travel expenses.

3.1.2. *Temporal Distance*

The difference of time zones between remotely located colleagues reduces the possibility of synchronous communication and is a cause of *delayed communication* which can result delayed execution of some tasks. Indeed, if a developer needs to obtain some information from a distant colleague in order to realise his/her task, he/she will probably need to wait the next morning to obtain the information (synchronous or asynchronous) and to finish the concerned task. Moreover, we have observed that in some cases when the required remote person was not available the communication was abandoned and led to *not transmitting of information* at all, especially with the intention of preventing the delay of some tasks.

Because of the temporal distance, *use of asynchronous communication increases*. This type of communication highly depends on the communication tools available for the project. We can identify some positive aspects in using e-mail, fax and other communication tools as for example keeping the trace of the exchanged documents and decisions. But, there are also some negative aspects, as for example, the necessity to formalise the exchanged information: ideas, questions, responses. It is necessary to guarantee that the transmitted idea will be well understood by the remote colleague in order to avoid needless work. It is evident that temporal distance and asynchronous communication increase the *delay of the feedback* especially if it involves several distributed teams.

3.1.3. *Socio-cultural Distance*

The language used for the communication with remote sites (generally English) can be a *barrier for synchronous communications* with not-native speakers which can have some difficulty to communicate in real-time and especially with a group of people. Indeed, these people prefer asynchronous communication which allows to formulate and verify their ideas before communicating them. Besides, the cultural distance can lead to *misunderstanding and different interpretation and representation* of a problem even though all participants speak the same language.

Cultural distance means also a difference in team management. For example, when several *synchronisation meetings* are planed in collocated teams, the remote teams may

do not have this habit and organise meetings only when an important problem is identified.

3.1.4. *Organisational Distance*

Organisation of teams is modified while integrating a distributed project and typical communication channels used within a collocated team are not adequate anymore. Changes in communication ways can entail a *non respect of information flows*. It can be caused by the lack of information about the rules that information flows have to follow in each organisation. For example, we have observed a phenomenon that when sending e-mails some people put all the addressees in the field "Carbon copy" because of their ignorance to whom the message should be addressed. Lindqvist et al. (2006) suggest to *define project communication channels and to pre-establish project priories* in order avoid the loss of time and superfluous communications. Scott et al. (2005) propose a set of *communication patterns* for improving communication in DISD projects.

3.1.5. *Technological Distance*

Because of the large offer of communication tools it is frequent that different partners use different tools. Having, for example, several e-mail addresses can cause some trouble for partners to choose the right address and sometimes it can lead to a loss of messages and bad coordination. Installing a *common e-mail server* for the project could be a solution providing that all project members use only this particular e-mail address for the project communications.

Event though the communication technologies (especially voice communication) have considerably evolved these last years, we can still experience interruptions or bad quality of sound that can hinder *communication quality*.

3.1.6. *Knowledge Distance*

Lack of language knowledge, in particular English, can be the origin of serious understanding problems during communications. Besides, difficulties can appear because of the *lack of specific knowledge* (e.g. application domain, technology, programming language) and could be a barrier to a good communication and lead to information misunderstanding.

3.2. *Coordination in DISD*

3.2.1. *Geographical Distance*

Lack of experience in working together as well as lack of face-to-face interactions can *reduce trust* between remote partners (Pyysiäinen, 2003). Kiel (2003) stresses that at a great distance it is difficult to empathise with people that you nether meet and it is easy to ignore them and devaluate their contributions and abilities. Moreover, this lack of trust can reduce people's determination to communicate with distant colleagues and to do

some development changes without consulting a necessary remote person (Bernard, 2006) and in this way to cause coordination problems.

The feeling to belong to a team can be reduced because of the geographical distance and lack of informal communication (Herbsleb and Mockus, 2003; Kiel, 2003). Because of the *lack of team spirit* the coherence of a group of people working together can be corrupted and cause difficulties in coordinating their work.

The smallest remote teams can experience the *lack of awareness* about the real project situation and the priorities of the clients. Indeed, it is easier to be aware of the changes made every day when all people are collocated. Lack of information can lead to a bad coordination between different project sites.

In order to increase clarity of communication, Lindqvist et al. (2006) advise to take *minutes of all meetings* even though this measure is not sufficient because of the various important things said before and after the meeting as informal communications. Another solution would be to *establish a system dedicated to share files and information*.

3.2.2. *Temporal Distance*

Because of the time zone difference the number of *overlapping working hours is reduced* between sites (Espinosa and Carmel, 2003; Kiel, 2003) and makes the coordination of collaborative work problematic. Even one or two hours time zone difference can considerable reduce the overlapping hours during a working day because of different beginning and end of the working day and different lunch time at different sites. Besides, it can be *difficult to organise synchronous team meetings* between remote sites. Some people might have to adapt their working hours in order to participate in such a coordination meeting (Battin et al., 2001). Finally, the *availability of technical infrastructures* used for remote teams' coordination can be problematic for the reason that most of the management tools do not allow 24/24h and 7/7 day access. Some periods of time are necessary for systems back-ups and synchronisations (Ebert and De Neve, 2001).

3.2.3. *Socio-cultural Distance*

Time used for taking a decision or an initiative depends very much on people's culture and on some sites it can be always longer than of other sites. The lack of initiative in taking a responsibility of some people can sometimes necessitate a person from a distant site to coordinate the decision making process. Besides, according to the cultural traditions the decision taking has to follow some hierarchical process which can be different from other sites. The *perception of this hierarchy* can pose a problem when coordinating work on different sites if for example questions or decisions are not addressed to the right person.

Cultural difference concerns not only people but also enterprises. When *enterprise culture* is not assimilated and understood in the same manner on different sites, the coordination of work can undergo some difficulties. Diverse *values and norms can vary* from one enterprise to another and therefore can be a source of difficulties. For example,

very often developers experience some frustration when they have to understand and/or modify the code developed by a distant team. Despite the same knowledge of the programming language, the cultural and development approach difference can lead to a different type of code and create conflicts between developers.

Specific application domain knowledge is generally required to realise projects. Organisations from different countries can have *incompatible view on some particular application domain*, based on their own experience and expertise (Curtis et al., 1988). We have observed that even though the developer from India was knowledgeable in the financial domain when developing an interface for an invoicing process he was not aware of Swiss taxation system.

Finally, the developers are often *doubtful of the capabilities* and skills of the remote teams (Battin et al., 2001) and have a difficulty to delegate them to some critical tasks. This situation can lead to the overload of work on one site.

To better coordinate work on several distant sites, Kruchten (2004) proposes to identify multicultural talented individuals able to operate in two different cultures and to use them as *"proxies"*. Because of their multicultural competencies and experience in distributed development projects, these people can hold leader positions and coordinate work on different remote sites.

3.2.4. *Organisational Distance*

Lack of organisational visibility, i.e. knowledge about people's skills and competencies, knowledge about communication channels and organisation hierarchy, can be a serious barrier for the project coordination.

The nature of DISD invites development teams to divide their work into well defined modules (Bass and Paulish, 2004). In this way, most of the decisions can be made on the modules and not on the whole project. However, the *interdependent nature of modules* can pose problems when integrating them into one system. Battin et al. (2001) suggest *incremental integration* of different modules in order to reduce the importance of the final integration. Besides, the *interdependency of the development processes* on different project sites is not always taken into account. The development process can vary from one site to another because of the lack of standard activities. This can lead to some incoherencies in working methods and delivery of the final product.

Because of the organisational distance, each project site generally has a project manager. As each of them attempts to manage the common project, *conflicts* concerning their role and responsibility in the whole project can emerge. This situation can generate a bad coordination and even duplication of work. On the contrary, if the person responsible for the project coordination on one site is missing for an important period of time, the coordination of work can be damaged as the people of remote site do not know to whom address their questions.

3.2.5. *Technological Distance*

Modularisation of work in DISD asks for the standardisation of the development environment, processes and methods. These standards, including manuals, databases and documents, are considered as key issues in work coordination (Sahay, 2003). *Lack of technology standardisation* can lead to bad work coordination and reduce work performance.

3.2.6. *Knowledge Distance*

DISD requires a *common understanding* of a problem or an entity between partners which can be difficult to obtain. Even though all partners accept to use the same standard (e.g. UML for system design) the knowledge of this standard on one site can be lower than on the other site and cause some misunderstanding. Furthermore, *lack of specific knowledge* required for realising a particular task, can make difficult the allocation of tasks to the organisations. Finally, some organisations are inclined to keep their expertise localized and do not wish to share the knowledge that brings them some identity. This kind of attitude can pose problems in work coordination; when the knowledge is kept too well, partners do not now about its existence at all.

3.3. *Control in DISD*

3.3.1. *Geographical Distance*

It is relatively easy to evaluate work progress situation and to be aware of work control problem when the whole project team is located on one site. Because of the geographical distance in distributed projects it is difficult to have a good *visibility of project progress* even though some project management tools are used. For example, we have met situations where developers from India assured that their work is progressing well while they were spending a number of additional hours to settle code errors. The difficulty was not transmitted to the general project manager in Switzerland because they considered the problem easy and fast to be corrected. This kind of problem would be identified very quickly on a collocated site.

The way that project managers lead distributed projects is different from the collocated ones. A *lack of involvement in project control* is perceived when project is distributed on several distant locations. Indeed, it is easier to ignore a problem when it appears on a remote site and to hope that it will resolve by itself. Some situations require much more than a simple communication and can ask the project manager to travel to the remote site. This effort is not always easy to realise (Lindqvist et al., 2006). We have observed that during the preparation of a distributed project very *few project planning meetings* were organised in comparison to the collocated projects.

Requirements changes are very frequent in IS development and obviously concern DISD projects. Sengupta et al. (2004) insist that *requirements change management* have to be visible to the all distributed project teams in order to avoid situations that some remote teams work with obsolete system specifications.

3.3.2. *Temporal Distance*

Because of the difference of the time zones, national holidays and vacations, *planning of tasks and resources* become very complex in DISD.

3.3.3. *Socio-cultural Distance*

The attitude with regards to distributed development varies from one project member to another. While some people see advantages in DISD others only expect problems and difficulties because of their past bad experience. Consequently, when *distributing tasks*, some project managers are reluctant to collaborate with remote sites and can cause conflicts of interest. Besides, the culture can cause important effects on the *interpretation and respect of the affected responsibilities and tasks*. For example, the developer can misunderstand why he/she didn't obtain some particular task. In this way, peoples' motivation and team management can be affected. Furthermore, the *authority can be perceived in a different way* according to the culture (Krishna et al., 2004) and the project manager has to take it into account while distributing tasks and responsibilities.

There is generally one main site in a DISD project which is responsible for the final product and thus parts of products developed on distant sites. However, it can be *difficult to motivate teams* that have no responsibility on products they develop (Lindqvist et al., 2006). *Interpretation and understanding of project priorities* can also vary form one project site to another. For example, we have observed that for the team from India the most important was to deliver their product in time even though the tests were not yet completed. A potential solution to this problem would be *to define project objectives, rules and priorities* at the beginning of the project, for example, to insist that "the quality is more important than the quantity".

Social and political norms (lows, traditions, regulations) of different countries have to be taken into account when planning DISD projects. For example, the time necessary to obtain a visa or a work-permit can vary according to the destination and the nationality of the person (Battin et al., 2001). The *interest in travelling* can also be perceived differently according to the person's cultural background and can pose problems when planning tasks requiring some people's delocalisation and travelling. Finally, it was observed that some remote sites *do not wish to share their knowledge and experience* with other DISD project partners in order to make themselves indispensable. The problems can arise when the project manager is not aware about all competencies hold by this site.

3.3.4. *Organisational Distance*

Working processes generally vary from one DISD project site to another and can cause some misunderstanding between partners. It was, for example, difficult to make understand the Indian team what they were supposed exactly to do during the test phase as their perception of this phase was different. It is important to learn about the expectations and workflows of other sites. We have observed that when providing two distant teams with the same requirements specifications, design specifications and

interface specifications, the obtained result was two totally different products. Lindqvist et al. (2006) highlight that *using the same method* on all DISD sites could help to resolve process misunderstanding problems.

Communication channels can vary according to the problem complexity and urgency. For example, an organisation can use vertical communication between different hierarchy levels to resolve some urgent problems or to establish rapidly some solutions but it is not appropriate when the remote sites are involved in the realisation of the concerned task. Finally, *management of distant resources* (human, budget) is not obvious because of the bad visibility of the real situation. Sometimes, travels have to be considered as necessary to obtain a better understanding of the situation on a remote site. In this case, the cost of travel has to be considered as minor in comparison with the resolved problems.

3.3.5. *Technological Distance*

The *lack of homogeneity between hardware and software resources* used on different DISD sites can cause problems when working of the same product (e.g. testing the same code on different platforms can give different results). This lack has to be minimised in order to guarantee common work control.

3.3.6. *Knowledge Distance*

Lack of the visibility of partners' skills and abilities can lead to a bad affectation of tasks and responsibilities.

3.4. *Development Activities in DISD*

3.4.1. *Geographical Distance*

DISD means that different parts of the IS can be developed on different sites. This leads to a difficulty in testing and correcting different modules because of their *interdependency*. Good *product decomposition* could minimise dependency of problems between geographically distributed sites (Pilatti et al., 2006).

3.4.2. *Temporal Distance*

Temporal distance has to be taken into account when testing different IS modules in order to avoid important loss of time. We have observed that a lot of time can be lost if difficulties and/or questions are communicated only the next day. If the temporal distance is not too big, it is recommended to *adjust the working hours* of the distant teams in order to maximise the working time overlap during some project periods.

3.4.3. *Socio-cultural Distance*

Teams with different cultural background can use different approaches for system analysis and design (e.g. different perception of abstraction, different architectural

patterns, etc.) even though they use the same modelling language (e.g. UML) and generate some difficulties during system design phase.

3.4.4. *Organisational Distance*

DISD requires *synchronisation of the development processes* of participating teams. Difficulties can arise when common processes are not understood well enough by some partners.

During the design phase, it is necessary to pay attention to the way the *product is decomposed with regards to the organisational decomposition*. This link can have important repercussion on the quality of the obtained results (Lindqvist et al., 2006) if the affectation of work was inefficient. Besides, in DISD each *modification in affecting resources to tasks* can be challenging especially when it is necessary to replace some exceptional expertise.

Development methods used on different sites can vary and pose problems during different development phases. For example, if one partner is in the habit to produce project documentation only in the end of the project and the other one aims to have it during the project according its evolution, conflicts can arise. Tests planed in the end of the development phase by one partner will not satisfy the other one waiting for iterative product validation and development. According to Lindqvist et al. (2006), a common method used on all DISD sites could help to avoid this kind of problems and improve project coordination and control. Besides, Paasivaara and Lassenius (2004) claim that *iterative and incremental development process* seems to be a viable approach for DISD.

It is common that user *requirements change* during the IS development. According to Sengupta et al. (2004), the difficulty is not only in the continuous necessity to modify code in order to meet requirements but also in ensuring that even if requirements change, a consistent view of the system is maintained across all the distributed project teams. For example, modifications done on one system module because of requirements change can have an effect on other modules. To better manage requirements change Sengupta et al. (2004) propose a *"test-driven global software development"* approach, i.e. to develop test cases for each new requirement and map the requirement to the test cases before any modification is made to the system code.

3.4.5. *Technological Distance*

The *difference of working platforms* used by DISD partners can induce various problems especially during the phase of tests. We have observed that code tested and validate on Indian partner servers was erroneous when tested on Swiss partners servers. Besides, the *difference of the database* used for tests can also generate problems or do not allow to reproduce errors obtained on partner's site. A potential solution would be to establish a *common centralised test infrastructure*. Bass and Paulish (2004) mention that technological factors can *limit design choices* to hardware, software, architecture, platform and standards that are currently available.

3.4.6. *Knowledge Distance*

Lack of knowledge on the application domain can cause bad specification of system requirements and/or bad design of the solution. Requirements specification can be considered by a distant partner as incomplete, insufficient or unclear while locally it is considered as well enough. Besides, misunderstanding problems can arise when one of the partners has *insufficient knowledge about the formalism* (e.g. UML) used for the system design. Finally, a good level of knowledge and specific ability can be required for each particular task in a project. We have noticed that knowledge gaps in one of the teams can have repercussion on the whole development chain. *Trainings* could be organised during the first phases of the project in order to obtain knowledge necessary for project realisation and to minimise the interdependencies between sites.

3.5. *Maintenance in DISD*

3.5.1. *Geographical Distance*

When a problem appears on the customer's location it is sometimes *difficult to send the right developer or consultant* to resolve it because this person is located on a geographically remote site. The travel expenses can also be of great importance. Big geographical distance between the customer and the development team make it *difficult to maintain a good level of support*. Indeed, the difficulty to have face-to-face communications can generate misunderstanding situations during the maintenance phase (Yan, 2004). Yan (2004) insists on the necessity to establish *agreements with the customer* in order to find a support solution in case of problems requiring a quick answer. This has to be considered during the contractual phase of the project. Besides, in order to guarantee the short time to answer the customer, Yan (2004) proposes to form and use *technical liaison engineers*. These persons should be qualified enough to handle most of the urgent maintenance problems and to reassure the customer.

3.5.2. *Temporal Distance*

Time zone difference between client's and developer's sites makes it difficult to have a prompt intervention in case of necessity for system maintenance. According to Yan (2004), it is important to set up *efficient communications* with the customer and to send enough *technical liaison engineers* to the customer's site for the local support.

3.5.3. *Socio-cultural Distance*

When maintenance and problem resolution take more time than expected the reaction of customer can depend on its cultural background and work routine. Some customers can accept the delay more easily than others and the lost of confidence can be observed. To avoid such a problem, Yan (2004) recommends to establish good relationships between csutomer's and development teams from the project beginning. Indeed, the good quality of these relationships facilitates the common comprehension.

3.5.4. *Organisational Distance*

Lack of standardised project documentation causes the difficulty for the long-term product maintenance. In fact, the development team evolves, people change, and if there is no good documentation of the developed products, standard formats of documents and models, it is difficult to guarantee efficient support and maintenance of the systems developed in the past. Standardisation of the DISD project management and organisation (Yan, 2004) is a difficult task but necessary to avoid various maintenance problems. Besides, it should go together with project members' training on formalisation of project documents, program code, etc.

3.5.5. *Technological Distance*

Difference of the technology used on different development sites and customer's site make it difficult to reproduce the same errors or problems point out by the customer, especially when the database used for the maintenance tests is not the same as customer's database. Indeed, some customers do not allow access to their database because of the evident security reasons. One of the solutions would be to establish *similar development and maintenance environment* between different development sites which also should be compatible with the customer's environment (Yan, 2004). Besides, a distant access to the customer's product system would help to react more rapidly. However, because of the high confidentiality reasons, and enterprise security policy, some enterprises do not allow such kind of access to their systems.

3.5.6. *Knowledge Distance*

Lack of knowledge about the final product can lead to system performance and maintenance difficulties. We have observed that very often the knowledge about the final product was distributed between several people. No one had a global and sufficiently detailed view of the system and no one was able to assure system maintenance alone.

4. Applying the Framework

As proposed in (Prikladnicki and Yamaguti, 2004), we think that it is indispensable to ask ourselves the right question before beginning a DISD project in order to anticipate potential risks and to avoid reproducing the same errors as in the past projects. We have observed that DISD projects can demand more time than the collocated ones as mentioned in (Herbsleb and Mockus, 2003). In fact, a lot of time can be lost during for example the phases of implementation and test. The reasons for that can be multiple: the difference of technological infrastructure, in particular different platforms and databases used for tests, not standardised code format, problems related to the coordination and integration of different modules and a bad visibility of the whole project progress. We applied our framework in a real project first to evaluate the degree of project distribution and to identify potential problems and then to select and establish some proposed solutions. In the following we present our study and summarise the results in Table 3.

4.1. *Context of the Project*

The enterprise of our case study is specialised in the development of financial IS (generic and customer-specific solutions) and is familiar not only with the IT domain but also with the financial world. The company is distributed on several sites between Switzerland, England and India with the headquarters located in Switzerland. While collaboration between Switzerland and England has more than ten years experience, the new development team was established in India only two years ago in order to extend enterprise skills and IT expertise and therefore to diversify provided services and solutions. A number of new problems came up from this new collaboration. These problems were corrected sporadically but the development speed imposed by the marked did not allow to really understand the reasons of the problems and to elaborate long-term solutions. It was decided to apply the framework during one of the last enterprise projects in order to improve DISD project management and to maximise the chance to succeed.

The objective of the project was to develop a financial solution providing user interface based on Internet technologies. Because of some development skills available only on the site of the Indian partners it was decided to realise a part of the development in India. Only some small modules were developed in Switzerland because of the required expertise in the financial domain. The test phases were realised by both teams and required a good coordination. All phases were managed from Switzerland.

4.2. *Determining the Risk of Distribution*

The risk of project distribution depends not only on the participating teams but also on the project characteristics and, for example, two projects realised by Swiss and Indian teams can have different distribution degrees because of different project requirements. Concerning this particular project, the risk of distribution was evaluated as follows:

- The *geographical distance* was considered as high mainly because of the difficulties related to travelling between the two sites. Even though the transportation offer is rather acceptable, it is not easy to travel between these two countries: the flight duration and cost are considerable and it is necessary to obtain a visa, which requires preparing the travel in advance.
- The *temporal distance* was moderate. In fact, four hours time zone difference can cause a big temporal distance if the required collaboration is very important. In this project, good decomposition into autonomous modules permitted to reduce the interdependency between sites. Only the phase of test had to be considered more seriously and endured some time loss.
- The *socio-cultural distance* was moderate because of the proxy person set up on the remote site as proposed in (Kruchten, 2004). However, the difference in the interpretation of project priorities was perceived.
- The *organisational distance* was big. The collaboration with India being rather new the organisational structure of the two sites was not homogenised yet: different working processes and different development methods.

- The *technological distance* was big even though some homogenisation of development tools and platforms was assured. However, during the test phase, a number of problems emerged because of the lack of standardisation, different test platforms and databases and difficulty to reproduce errors.
- The *knowledge distance* was moderate as the Indian developers had enough technical knowledge necessary for the system development. Of course, the project required specific knowledge related to the application domain, in this case financial, but it was decided to develop these specific modules in Switzerland. However, the minimal knowledge of the application domain was necessary and some knowledge transfer to India was indispensable.

Table 3. Application of the framework.

Activity	**Dimension**					
	Geographical Distance	**Temporal Distance**	**Socio-cultural Distance**	**Organisational Distance**	**Technological Distance**	**Knowledge Distance**
Communication	! Additional cost for travelling !Effort for travelling					
Coordination			* Proxy persons		! Lack of technology standardisation	
Control			!Interpretation of project priorities	! Different working processes *Common infrastructure for information sharing		
Development		! Loss of time *Adjustment of working hours		! Different development methods	! Difference of working platforms ! Difference of databases * Common test infrastructure	! Lack of domain-specific knowledge * Trainings
Maintenance					! Different technology for maintenance	

4.3. *Identifying and Establishing Solutions*

Based on the solutions proposed in the framework and the identified problems, some solutions were developed and established for this project. First of all, it was decided to homogenise the development platform, in particular by establishing a common environment for system tests. This environment was configured on the Swiss partner's network with an access from India via a VPN (Virtual Private Network) connection. This solution reduced the number of problems related to the use of different system and

database versions. It also decreased the loss of time due to the tentative to reproduce problems identified on other platforms.

The second decision was to adjust the working hours of the two sites especially during the test phase. In this way, the work overlap was increased and therefore system tests were better coordinated and realised more quickly.

In order to improve the project visibility, it was decided to establish a web-based portal for project information sharing. This portal was very helpful for exchanging project documents and permitted to have a global view of the project and to evaluate work progress of each participant. It also supported automatic versioning of the documents and offered a better traceability of the identified problems and solutions. Finally, the portal improved the communication about development progress and tests which had a positive effect on people's motivation and also was a gain of time.

As mentioned before, this project required knowledge related to the financial domain which was not available on the Indian site. Such a lack of domain-specific knowledge generally requires the separation of technical tests from the functional ones and can cause a big loss of time. It was decided at the beginning of the project to organise a training of the Indian team on the required financial knowledge. In this way, the interdependency between different tests was minimised and the Indian team was able to realise not only technical but also some simple functional tests. The most important functional tests were still realised in Switzerland.

5. Conclusion

The main challenges of DISD consist in overcoming the complexity and ensuring the quality of distributed development activities. Indeed, the activities of communication, coordination, project control, development and maintenance of the solution are affected when the development teams are spread out in several remote locations. Problems related to different DISD dimensions make the project management difficult and therefore increase the risk failure. The framework presented in this paper aims to characterise the domain of DISD; it has several purposes:

- It offers a structure for classifying DISD problems and solutions and contributes to the better understanding of the distributed development domain;
- It helps researchers and practitioners in identifying new problems and indicates which of them still need solutions;
- It can be used by the DISD project manager for better understanding the situation of the project at hand, identifying risks related to the development process distribution and establishing means to deal with the potential problems and risks.

Multiple causes of problems as well as various activities affected by the same problems make it difficult to establish such a classification. For example, we have noticed that geographical distance is the main cause of the lack of informal communication which can generate coordination and development problems of distributed projects. Coordination difficulties can in their turn create development problems like integration of modules developed on different sites. Such a complexity of

problems offers exciting research perspectives. It would be interesting for example, to examine problems in a more detailed way and identify their interdependencies and overlap zones.

The three dimension that we have added to the previous proposals (Agerfalk et al., 2005) aim to improve the framework and provide a better visibility of this complexity. We have noticed that technological, organisational and knowledge distance affect in a very important way different DISD activities, including development and maintenance of the IS product, and should be taken into account during the DISD project management. The framework gave us an opportunity to position our observations in real projects and better understand the complexity of the DISD domain.

References

Agerfalk, J., Fitzgerald, B., Holmstrom, H., Lings, B., Lundell, B., Conchuir, E. (2005): A framework for considering opportunities and threats in distributed software development. Int. Workshop on Distributed Software Engineering, Austrian Computer Society, pp. 47-61.

Bass, M., Paulish, D. (2004): Global Software Development Process Research at Siemens. 3rd Int. Workshop on Global Software Development (with ICSE 2004).

Battin, R.D., Crocker, R., Kreidler, J. and Subramanian, K. (2001): Leveraging resources in global software development. IEEE Software, Vol. 18, No. 2, pp. 70-77.

Bernard. (2006): Penser la mise à distance en formation. Apprendre à distance, Serres, Le Monde de L'Education, pp 88-90.

Boland, D., Fitzgerald, B. (2004): Transitioning from a Co-Located to a Globally-Distributed Software Development Team. 3rd International Workshop on Global Software Development.

Carmel, E. and Agarwal, R. (2001): Tactical approaches for alleviating distance in global software development. IEEE Software, 18(2), 22-29.

Casey, V., Richardson, I. (2006): Uncovering the Reality within Virtual Software Teams. Int. Workshop on Global Software Development for the practitioner, (with ICSE 2006).

Cherry, S., Robillard, P. (2004): Communication Problems in GSD: Spotlight on a New Field of Investigation. 3rd Int. Workshop on Global Software Development, (with ICSE 2004).

Curtis, B., Krasner, H. and Iscoe, N. (1988): A Field Study of the Software Design Process for Large Systems. Communications of the ACM, Vol. 31, No. 11, pp. 1268-1287.

Ebert, C. and De Neve, P. (2001): Surviving Global Software Development. IEEE Software, Vol. 18, No. 2, pp. 62-69.

Espinosa, A. and Carmel, E. (2003): The Impact of Time Separation on Coordination in Global Software Teams: a Conceptual Foundation. Software Process Improvement and Practice, Vol. 8, pp. 249-266.

Evaristo, J.R., Scudder, R., Desouza, K.C., Sato, O. (2004): A dimensional analysis of geographically distributed project teams: a case study. Journal of Engineering and Technology Management, 21(3), 175-189.

Grinter, R.E., Herbsleb, J.D. and Perry, D.E. (1999): The geography of coordination: Dealing with distance in R&D work. Int. Conference on Supporting Group Work, Coordination and Negotiation, p. 306-315.

Herbsleb, J.D. and Grinter, R.E. (1999): Splitting the Organization and Integrating the Code: Conway's Law Revisited. 21st Int. Conf. on Software Engineering (ICSE'99), ACM Press, pp. 85-95.

Herbsleb, J.D., Mockus, A. (2003): An Empirical Study of Speed and Communication in Globally Distributed Software Development. IEEE Transactions on Software Engineering, 29(6), pp. 481-494.

Kiel, L. (2003): Experiences in Distributed Development: A case Study. International Workshop on Global Software Development.

Krishna, S., Sahay, S. and Walsham, G. (2004): Managing Cross-Cultural Issues in Global Software Outsourcing. Communications of the ACM, 47(4) pp. 62-66.

Kruchten, P. (2004): Analyzing Intercultural Factors Affecting Global Software Development. The 3rd International Workshop on Global Software Development, (with ICSE 2004).

Lindqvist, E., Lundell, B., Lings, B. (2006): Distributed Development in an Intra-national, Intraorganisational Context: An Experience Report. First International Workshop on Global Software Development for the Practitioner (with ICSE 2006).

Malone, T.W. and Crowston, K. (1990): What is coordination theory and how can it help design cooperative work systems? ACM conference on Computer-supported cooperative work CSCW '90. ACM Press.

Malone, T.W. and Crowston, K. (1994): The interdisciplinary study of coordination. ACM Computing Surveys, 26(1), pp. 87-119.

Mcchesney, I.R. and Gallagher, S. (2004): Communication and co-ordination practices in software engineering projects. Information and Software Technology, 46(7), pp. 473-489.

Nissen, H.W. (2004): Designing the Inter-Organizational Software Engineering Cooperation: An Experience Report. 3rd Int.Workshop on Global Software Development (with ICSE 2004).

Paasivaara, M. (2003): Communication Needs, Practices and Supporting Structures in Global Inter-Organizational Software Development Projects. International Workshop on Global Software Development (with ICSE 2003).

Paasivaara, M. and Lassenius, C. (2004): Using Iterative and Incremental Process in Global Software Development. 3rd International Workshop on Global Software Development.

Pilatti, L., Audy, J., Prikladnicki, R. (2006): Software Configuration Management over a Global Software Development Environment: Lessons Learned from a Case Study. First Int. Workshop on Global Software Development for the practitioner, (with ICSE 2006).

Prikladnicki, R., Yamaguti, M. (2004): Risk Management in Global Software Development: A Position Paper. 3rd Int. Workshop on Global Software Development (with ICSE 2004).

Pyysiäinen, J. (2003): Building Trust in Global Inter-Organizational Software Development Projects; Problems and Practices. Int. Workshop on Global Software Development (with ICSE 2003).

Robillard, P.N. (1999): The Role of Knowledge in Software. Communications of the ACM, 42:1, 1999, pp. 87-92.

Robillard, P.N. and Robillard, M.P. (2000): Types of Collaborative Work in Software Engineering. Journal of System and Software, 53, 2000, pp. 219-224.

Sahay, S. (2003): Global software alliance: the challenge of "standardization". Scandinavian Journal of Information Systems, 15, pp. 3-21.

Scott, A., Izquierdo, L., Gupta, S., Elves, R. and Damian, D. (2005): Leveraging Design Patterns in Global Software Development: A Proposal for a GSD Communication Pattern Language. International Workshop on Distributed Software Development (with IEEE RE'05).

Sengupta, B., Sinha, V., Satish, C. (2004): Test-Driven Global Software Development. 3rd International Workshop on Global Software Development (with ICSE 2004).

Wiredu, G.O. (2006): A Framework for the Analysis of Coordination in Global Software Development. First International Workshop on Global Software Development for the practitioner (with ICSE 2006).

Wiredu, G.O. (2005): Coordination as the Challenge of Distributed Software Development. International Workshop on Distributed Software Development (with IEEE RE'05).

Yan, Z. (2004): Efficient Maintenance Support in Offshore Software Development: a Case Study on a Global ECommerce Project. 3rd International Workshop on Global Software Development (with ICSE 2004).