Distributed Development in an Intra-national, Intra-organisational Context: An Experience Report

Edvin Lindqvist
University of Skövde
P.O. Box 408
SE-541 28 Skövde, Sweden
+46 (0)500 - 448000

edvin.lindqvist@gmail.com

Björn Lundell
University of Skövde
P.O. Box 408
SE-541 28 Skövde, Sweden
+46 (0)500 - 448319

bjorn.lundell@his.se

Brian Lings
University of Skövde
P.O. Box 408
SE-541 28 Skövde, Sweden
+46 (0)500 - 448000

brian.lings@his.se

ABSTRACT

This paper reports from a study investigating distributed development in an intra-national and intra-organisational company context. The study gives an insight into issues related to, and strategies for successful distributed development seen in a project at Ericsson Microwave Systems. We conclude that regular, informal communication is the single most important factor for success in the project, with important roles in improving motivation and coordination. The leader's role in setting clear and transparent project priorities, and the use of iterative development methods with regular deliverables between sites, are identified as key elements of success. The context of the study is a relatively undemanding but common distributed development context, and the issues raised and strategies found to be effective are likely to be relevant to broader distributed development enterprises.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management - *Programming teams*

General Terms

Management, Human Factors

Keywords

Distributed Software Development; Distances in development; Geographical distance; Socio-cultural distance; Communication, Coordination, and Control in life-cycle activities.

INTRODUCTION

Software development has not been excluded from the current trend of increasing industrial globalization. It is no longer unusual for development teams in the same project to be located in different cities, or even in different countries. There are several factors behind this trend, including a need to reduce costs, increase competition, and use resources from many different geographical locations [13]. The trend is not only for development to be distributed over national borders, but over organizational borders as well. A steady increase in outsourcing is one aspect of this trend [11].

Utilising geographically distributed teams does not always increase efficiency; in fact the opposite is often the case. Herbsleb and Mockus [5] indicate that time requirements for distributed development (DD) are often more than double those for localized development. According to Herbsleb and Mockus, the main reasons for this relate to difficulties with communication and coordination.

The available research on DD to a large extent deals with the issue of DD in an international context, sometimes referred to as Global Software Development (GSD), see for example [2] [3] [6] [7]. There are also several studies focussing on DD in an inter-organisational context (e.g. [4] [9] [11] [12] [14]). However, it is important to recognise that DD also exists in national and intra-organisational contexts. This kind of DD might even be the most common. It is therefore important to consider such baseline cases, which will reflect the experience of many practitioners within the EU and can give insight into the core issues of DD.

The paper is structured in the following way. Section 2 gives an overview of how the study was carried out. Section 3 gives a presentation of the organisation under study. Section 4 presents the empirical data. Section 5 contains an analysis of the data, presenting both issues related to DD and strategies for successful DD. Section 6 includes a discussion and conclusions.

RESEARCH METHOD

An initial literature review was used to identify the most important issues for capturing an interesting and realistic representation of the work in the project studied. The following areas were seen as most important, and created the foundation for the interview guide:

The project. The project must be described in enough detail to assess "distributedness" and generalisability.

Product structure vs. organisational structure. The division of the product over the organisation might have an impact on the result.

Power. An unequal division of power might lead to distrust and other problems.

Communication. Since communication often is a great creator of problems in DD it is important to see how communication is used.

Ways of working. Methodological work practices might have an important effect in the distributed environment.

Organisational culture. Values in the organisation might affect the possibility of success in accomplishing DD.

Interviews were used for data collection. The interview guide[1] contains both pre-phrased questions and a number of relatively open areas to be further elaborated during the interviews. The interview guide was not strictly followed; in fact wide digression was allowed. However, the interviewer placed great importance on addressing all pre-specified issues. Conversations held and notes taken during all interviews, including all questions and answers, were held in the native language (Swedish) of the interviewer and all interviewees. The interviews were recorded using a tape recorder, and later transcribed in their entirety, resulting in about 30 pages of transcription. A total of three interviews were conducted, with participants from the two sites involved in the project. When selecting interviewees, care was taken to find individuals with long involvement in the project.

To get as complete a picture of the project as possible, including identifying issues on different levels, interviewees with different roles were used:

- A developer

- A team leader

- A manager / sub-project manager

Analysis was conducted using pattern matching [15], i.e. trying to find similarities in the data. This made it possible to identify a number of issues with DD encountered in the project, and a number of strategies used for managing these issues. The transcribed data was translated into English only after analysis, and for the purpose of reporting the work.

We relate the current work to the broader context of GSD by using an existing framework [1] to present our results. This framework categorizes issues with DD from two viewpoints: distance [temporal, geographical, socio-cultural] and process [communication, coordination, control] (see Table 1).

Table 1. Categories used for presentation of data

	Temporal distance	Geographical distance	Socio-cultural distance
Communication			
Coordination			
Control			

As observed in [10], distance can be experienced in a dimension for a variety of reasons, even within an intra-national, intra-organisational project. It is not only apparent when projects are globally distributed. The framework was not used as an a priori structuring device in the research design.

The interviewer had previously been employed at Ericsson Microwave Systems and worked in the project under study, primarily in the role of team-leader at the Skövde site, and in requirements engineering. This gives the results additional strength since the initial insight into the project and organisation was deeper, and the terminology well known. However, the potential weakness of having an internal interviewer, for example guarded responses because of internal power structures, were mitigated because the interviewer was no longer with the company but had maintained good relationships since leaving.

CONTEXT OF THE STUDY

Ericsson Microwave Systems is part of the Ericsson AB telecom corporation. The company is divided over several sites both within Sweden and internationally. The largest and dominant site is located in Gothenburg, with about 1600 employees. Most projects are accomplished in a co-located environment at this site.

The project under study is fairly large, involving in total more than 100 persons. It is located at two Swedish sites, Gothenburg and Skövde. The Skövde site has about 30 employees, of which seven were working with the project in its most intense phases.

The development method used in the project, MOOSE, is an internally development method based on Objectory (the RUP predecessor) [8] and has been in use since 1993.

The development is supported by a number of CASE-tools. Some of these can be seen as important enablers of distributed development. The following tools enable all developers to work with the same artefacts or help the distributed development in some other way: Rational ClearCase (configuration management and distributed file system), Telelogic TAU UML Suite (visual modelling), Rational ClearDDTS (issue/defect management), Serena RTM (Requirements Management).

The project was in the delivery phase when the study was conducted. Interviews were conducted at both the Gothenburg and Skövde sites. All interviewees had worked together with DD in other projects in the organisation prior to this project. The level of distribution of the project can be categorised as follows.

Temporal distance is low. There is no time-zone difference between the sites. The co-workers have flexible work times, but are recommended to be available at the office between 8:30 and 15:30. Time for lunch varies, but most have a 40 minute lunch break somewhere between 11:00 and 13:00.

Geographical distance is moderate. The two sites are located about 150 km apart. Most co-workers use rail for travelling between the sites. A one-way trip by train takes slightly more than an hour. Travelling by car takes slightly less than two hours. The minimum time needed for a trip is half a day.

Socio-cultural distance is low. The Gothenburg site has a long history, while the Skövde site was founded about eight years ago. A culture quite similar to that in Gothenburg has been assimilated by the Skövde site.

The Skövde site has assignments with different departments of the Gothenburg site, while the co-workers in Gothenburg mostly work on projects from their own department. The co-workers in

---

Skövde can thus face slightly different cultures and work practices in different projects with Gothenburg.

EFFECTS OF DISTANCE

This section presents the data gathered during the interviews, using the structure of the previously described framework. Some parts of the framework had no significance for the described case, and are thus not included.

Temporal Distance

Even though the sites are located in the same time zone, some issues are seen which resemble those experienced when there is a temporal distance between sites.

Communication

The interviewees prefer to phone rather than use mail for communication with the other site. This is to avoid the unnecessary delays often experienced with asynchronous communication. Misunderstandings are also seen as more common when communicating by mail.

One of the interviewees has noticed that developers unaccustomed to DD often use mail when communicating with the remote site. The same developer confesses that the likelihood of using mail instead of phoning is greater if the person on the opposite side is unknown, even given the awareness that this might create problems. The use of different means of communication is illustrated below:

"Most people mail me. I prefer to call. I believe it is easier to call. When someone mails me I pick up the phone."

Geographical Distance

The distance between the sites seems to be what creates most problems in the development. One of the main reasons for this is limited opportunity for informal communication.

Communication

The following quotation illustrates both the challenge of communicating and the need for communication in a distributed environment:

"I've seen communication fail several times when people haven't communicated when needed. It's enough to call once a week, send a mail with the current status, or anything and it will work out alright. The means are simple, but still people don't bother or don't believe it's important. It's enough to ask 'Hey, what's up today?' It's important to make the call, even if you don't have anything important to say today. You have to work with it; you have to show up at the different sites. In the same way as developers must travel to their colleagues at the main site often, something that is probably troublesome if you have a family and so on, the managers and project leaders must travel to the sites where the work is done. Preferably some of the co-workers as well. It doesn't hurt to travel for attending reviews and project meetings."

As shown in the quotation above, it is often the developers from the small sites that travel to the larger site, but that is not enough. Managers and project leaders must travel to fulfil their responsibilities. Travel allows other key people to get a better feeling for what is going on at the collaborating sites.

There are seldom problems with gaining permission to travel from the leaders in the organisation when it is necessary. Instead it is the developers that minimise travel. Issues requiring travelling are often considered to be less important, despite the fact that a possible resulting delay in the project is far more expensive than the cost of travel. The problem is illustrated below:

"To take a short meeting, or take a seat in someone's room when you have a problem is something you do all the time. But the problem needs to be at least ten times worse to order a train ticket. ... A train ticket is nothing compared to a delay in the project. It's not even in the same galaxy of costs."

The problem can also be a lack of willingness to travel. It is often much more convenient to stay at home. The culture described above applies foremost to Gothenburg. The Skövde site is dependent on travel to justify its existence and has had a culture of travelling from the start, but the description is still to some degree valid for Skövde. A deeper insight into the travelling culture in the company is given below:

We don't have the culture to have synchronization meetings without having something important to discuss. So, when problems emerge, they always come as emergencies. An event must trigger the meeting."

Coordination

Even though ongoing development probably demands most coordination effort, the most important coordination dependence of the smaller site in Skövde on the one in Gothenburg is to be updated on the needs and priorities of the customer.

Communication is said to be really important to maintain coordination. If communication fails it can cause serious trouble, as illustrated below:

"The worst case is that it doesn't work at all. If it's located at one site something will always work. Everyone has a task, so if you are sitting pretty close to each other and attend the same meeting you will get similar information, and you will always get something. But if the communication between two sites does not work you won't get anything. Different interfaces are used, different ways of working, completely different functions are implemented etc. There's no end to how completely wrong it can get."

The interviewees say that it is not enough just to communicate; the quality and clarity of the communication must increase when working with DD. One way that was mentioned is to use minutes of meetings. However, this is still not enough since much is said before and after a meeting which is also important:

"The first five to ten minutes when people are gathering and the last five to ten when they are leaving, if they aren't in a hurry for the next meeting, there is as much communication as during the meeting."

The co-workers at the Skövde site feel that much of the informal communication that highlights what is of immediate importance is missing. At the same time this can also work to their advantage. Since they do not share the panic when a major problem is discovered they work steadily towards their goal, without putting too much effort into problems that sometimes are exaggerated.

This has resulted in a cost more in line with what was initially estimated, i.e. cheaper than may otherwise have resulted.

Even though the sites have a long-term relationship the need to meet occasionally is identified as important. The importance is illustrated below:

"If I had only spoken with someone over the phone for ten weeks I wouldn't have understood the person as well as if I had looked him in the eyes once every two weeks and could see and understand what was really important and what was only done because of informal rules. If you meet a person you always get a little more information than over the phone. You speak a little nonsense and he remembers 'It was this thing also…' This part is really important."

The ability to share artefacts and information in CASE-tools was seen as very helpful. One of the advantages is illustrated below:

"When you are coding you must have all information necessary to do your job. Apart from that there is no difference whether I'm sitting in Skövde or Gothenburg since we have a shared file system with configuration management."

Control

Managers and project leaders often do not work in the same way with remote sites as with their local site. It is very easy to ignore a potential problem, even if you have a feeling that there are troubles at the remote site. Instead you hope it will be solved without intervention, because it is a little harder to grasp what is going on there. The trouble is that such problems seldom go away. An example of the opposite behaviour, representing excellent DD leadership, is illustrated below:

"I try to keep updated of what's going on. When I notice that something isn't as I want it to be I make a phone call immediately and speak with them (the developers) in person. I never send mail or such things. That's a later issue. Preferably, I walk up and speak with them. But you can't just walk up to Skövde on a Tuesday afternoon; then you have to call them. Contact with the individual is really important."

While some managers and project leaders understand the need to react immediately to problems, and this is also true at the remote site, there is very little long-term management between the sites. Action is taken only when something happens. One of the interviewees gave the following tip about how to act as a leader in an organisation using DD:

"You have to be consistent in your leadership methods, irrespective of whether it's Luleå, Skövde or Mölndal. If your way of leading works in Mölndal, you can be damn sure it will work well also in Skövde."

One issue discussed in the interviews was development methods. The interviewees were agreed that what is most important is not which method is used, but that the *same* method is used at all sites. It is very important to know what to expect from the other sites, that deliverables use the right version of interfaces etc.

The interviewees were also agreed that some methodological techniques give additional advantages. For example, the use of iterations in the development process will enforce communication, since the developers must communicate at least at the beginning and end of each iteration. This is illustrated below:

"You take smaller development steps one at a time and check that you've done it right at the end of each iteration. It's good to work in this way, since the risk is smaller compared with working with different tasks for a long time."

But only testing at the end of iterations is said not to be enough; you have to test from the start. This is illustrated below:

"There doesn't have to be anything at all behind. Just test that the communication works and that we can connect. Don't do it last, but begin with it. If it's possible to test, we should test it. We have always had verification as something that should be done last, and that doesn't work. The tester should start testing at day two, day two after the coder has started coding."

The method formalizes communication and coordination. It is important to remember that it is not enough to only trust to methodological, formalized communication. The possible effect is described below:

"You create a design specification, a requirements specification, and an interface specification. Then both sites start working, following the documents describing the coming increment. If they don't speak to each other they will end up with two completely different products… And you have a month of lost work. This has happened in several projects."

Vertical communication, i.e. communication between different hierarchical levels in the organisation, is primarily used when priorities must be set quickly and when there are big problems. These channels of communication are consequently not used and maintained to the same degree as horizontal channels. This can be troublesome since, when problems arise, they must often be resolved quickly. To manage this problem it can be useful to have priorities set between projects. It is also important to have clear roles to make communication channels visible. This is illustrated below:

"… it would be an advantage for everyone to know that this has priority X. You wouldn't have to go through the organisation for a day or two just to get hold of the project leader you need to discuss the prioritisation with."

Socio-Cultural Distance

All sites in the project are part of the same company and are located in the same nation. Organisational and national cultures are therefore congruent. Socio-cultural issues are therefore limited, but still exist.

Communication

Attitudes towards DD were divided. Some perceived DD to be necessary but to create many problems; others perceived DD to offer advantages over co-located development. An example of the latter can be seen below:

"It is possible to achieve positive effects if you use it in the right way. You can use it to gain a momentum in a development project. By creating interdependencies, it can grow to something bigger. … Most common is that people don't have enough time or aren't good enough leaders, which will cause problems, because it requires much more effort to get such co-operation to work."

An insight into how to achieve these added values is described below:

"If you manage the communication, and make sure the sites keep in close contact, and travel to the different sites, schedule meetings and so on, and make sure they are dependent on inputs/outputs to/from each other, then I believe it is possible for the motivation to increase and you can get a momentum in the project... You can achieve blast-off. You can get a feeling of competition between the sites as well, like now when we are correcting bug reports. And you can pep up the others. Both helping each other, and competing a little."

When new co-workers enter the project at the Skövde site, the team leader takes responsibility for their contact with the other site. The reason is that it will take too much time to create a social network from the start. The more experienced project members manage most of their communication by themselves, since misunderstandings are more likely to occur with an intermediary. Reasons for using an intermediary are presented below:

"They feel unsure about their part, and then you have to join in because there is no time, there isn't calendar time for social activities like kick-offs or such. The consequence might be that I have to take part. It's ok with the double man hours, calendar time is more important."

Coordination
The smaller sites in the company are eager to create their own profile. This is a way to gain a *raison d'être*. To achieve a competitive advantage the site can keep unique knowledge to itself. The problem is that nobody will even know that the knowledge exists and it will consequently be used to a smaller degree, or even not at all. This is illustrated below:

"They want to explore new areas; they want responsibilities of their own. They make a great effort with the local universities, supervising theses and such things. But the information gets stuck at the sites."

According to one of the interviewees, one good way to disseminate expert knowledge from the sites is to arrange recurrent meetings with managers from the different sites and inform them on what is going on in the projects, in cooperation with universities etc.

Another important issue raised is that it is not certain that cooperation would have been more successful if both sites were located closer than 150 km apart, or even in the same town. The main factor behind cooperation working well in most cases is the long running relationship between the sites.

Control
It is often harder to motivate an individual employee to make a short and temporary contribution to a project than it is to get an agreement with managers and project leaders:

"It is much less trouble with that part, to increase the priority of a project from the organisation or managers, than it is to motivate an individual co-worker that he must switch projects for a week or two."

The developer sees his current project as important, and by not working with it for a couple of weeks the timetable will slip and milestones may be missed. Motivating a change of project can be particularly hard at remote sites since the urgency and panic in the project does not show as clearly as at the main site.

A temporary change also puts more strain on a developer at a remote site since the channels for communication to the main site that may have existed earlier may no longer be very active. A solution to the motivation problem is for the leader to put additional effort into making what is prioritised clear.

The main site of the company often owns and has responsibility for the entire product, including the parts developed by the smaller sites, as in the case studied. One reason is that there is no culture of sharing ownership; an attitude persists that the responsibility should be safeguarded in the interest of the main site. Motivation is often increased by owning what you are working on. Such motivation can increase in the work in general, but it can also be easier to motivate co-workers to temporarily change project.

"Above all, I believe it would be possible to gain motivation from the feeling of 'Wow, it's ours!' You get a completely different verve and it is much easier to internally motivate why we should lift X from this work when we have problems. We will suddenly decrease the number of people necessary to reach agreement with."

The different sites are eager to take responsibility for and further develop the product they have created. This is something they rarely get money for automatically. The product leaders on the main site responsible for project resources do not have ready insight into the products developed on the remote sites, and as a consequence no greater interest in their further development. They are more motivated to develop the locally developed parts that they know well. This means that the sites are forced to ignore and pass over the product leaders: something that can result in both losing the overall picture and making the product leaders upset. The reason for this is illustrated below:

"They (the developer) want to have control over the tasks, and they (the product owner) want to have control over the product"

All interviewees were under the impression that it would have been better if the Skövde site had had primary responsibility for the product, rather than just working on its development.

ANALYSIS
The analysis will further elaborate on the empirical data by identifying a number of issues creating problems in the project, and strategies used which contributed to successful working with DD.

Issues From DD in The Case
An important issue seen in the culture of the organisation is the recognition of the need for *communication and travel*. The general pattern of behaviour implies that communication by phone and travel almost only occur when there is something of real value to discuss, or when problems emerge. This is not enough. More continuity is needed in communication, which needs to be of a higher quality if the success of the project is not to be jeopardized.

A lack of continuity in communication and of informal communication makes is hard at remote sites to see what is

important. This leads to an *underestimation of problems* at remote sites. This makes developers even less willing to travel. The probable consequence of this is late problem discovery involving increased costs.

The problem described above can also be seen in leadership, where the leader often has a *different style of leadership* between the local and remote sites. Also the leader puts more focus on local problems. Most leaders react immediately when they feel something is not quite right at the local site, but often let this feeling pass by when it concerns a remote site. Even if some leaders manage this issue, almost all have problems with managing *long-term leadership* of the remote site.

Trips are, to a large degree, made by the developers from the smaller sites, making the problem of *remote site awareness* even more serious for leaders and developers on the main site. This also leads to less awareness and use of knowledge related to the profile of the smaller site. This lack of awareness might also create motivational problems when a developer on a small site has to change project for a short period. It is harder to be motivated when you do not have a feeling for the urgency in the other project, and see only the delay in your current project.

Co-workers unaccustomed to DD often use mail for communication with remote sites. However, the use of *asynchronous tools* such as mail creates delays in communication. The experienced are well aware of this and primarily use the phone. Even so, there is a tendency to use mail when communicating with someone unknown, even for developers experienced with DD.

The main site is often *owner of the product* developed by the smaller sites. This probably affects the motivation of the developers. You are often more motivated when working with something you own, than doing it for someone else. The ownership also creates problems when planning for further development. The small site often has a desire to have some control over the development and plan for further development, but the comprehensive picture might then be lost. The main site, on the other hand, does not have a deep knowledge of the product, and might not be eager to plan for further development.

The use of a development method is important, but it might also be a creator of problems. If only *formal communication from the method* is used, there is a great risk of creating two completely different products on two sites, since it is hard to understand the thoughts of someone you do not talk with.

Strategies for Successful DD

It is important for communication between the sites that everyone involved in the project travels to the remote sites. Not just developers, but project leaders and managers as well. Travel should not only be made when problems and important issues emerge, but on a more regular basis. This makes it easier to disseminate expert knowledge between sites and gives a better awareness of what is happening on other sites. Even so it is important to realize that problems on the other sites are probably just as serious as the local ones, and should not be ignored.

However, a lower awareness of what is going on at other sites might sometimes be an advantage. Errors leading to panic are sometimes much less serious than first believed. Without getting the informal signals that something has to be done right now a developer will focus entirely on the problem at hand and continue to work steadily. This is an advantage when the problems are eventually revealed to be minor.

Communication in a DD environment needs greater clarity and higher quality. One way to increase clarity is to take minutes of meetings. This gives the possibility to better recall decisions made, something often made by a verifying question at the coffee table in a co-located environment. By using distributed CASE tools off a common repository, the developers can access and work with the most fundamental artefacts from all sites. Most communication between the sites is made by phone. But this is not enough. To really understand the person on the other end you have to meet now and then.

Vertical communication is almost only used when there are problems, or priorities have to be decided. Vertical communication channels are thus probably not very efficient. Since issues raised using vertical communication often need quick decisions it is important to have clear roles. Another way is to have pre-established project priorities.

The manager working in a DD environment must be aware of the management style used towards both the local and remote co-workers. The best way for the manager to work is probably by using the same management style for both sites, even if it takes a little more effort. The following might help a manager to engender success in a DD project:

- Create interdependencies between sites

- Make sure co-workers at the different sites keep in contact by encouraging communication and scheduling meetings.

- Create a sound feeling of competition between the sites.

When there is a need to make a temporary move of a co-worker between projects it is important to be very clear in describing the reasons for the move, to avoid losing motivation. Having a designated contact person might also help in getting started more quickly. The same is true when a new member enters the project.

The motivation at smaller sites might increase if they own and are responsible for the product they develop. This might also make planning of further development easier. The developing sites should in any case take part in the planning of further development.

When a development method is used in a DD project it is important that all sites use the same method, since the method sets up a framework for communication. Other ways in which the method might help communication is by using an iterative way of working and a continuous testing approach.

DISCUSSION & CONCLUSION

Distributed development can both be a creator of added value and a source of problems. In this study the most important factor for success or failure in DD is how well communication works. Effective communication can create highly motivated employees using the combined experience from all sites, while a complete communication failure probably leads to large additional costs.

One of the most important roles for creating a culture of good communication is the leader. Irrespective of whether it is a

manager, project leader or team leader, someone must motivate the co-workers to put in the additional effort required for effective communication in a distributed environment. The leader is also best placed to effect an environment supportive to DD.

The findings from our observations in this study are potentially relevant to any similar environment and development scenario, i.e. an intra-organisational, intra-national DD environment. Since the communication barriers are low in such an environment compared with, for example, GSD the identified *strategies* might be hard to generalize outside this domain. The *issues* on the other hand are more likely to be relevant outside this domain. This study identifies issues in a very *supportive* environment for DD (same language, culture and organisation) and thereby shows the problems that first emerge. The broader DD domain is likely to have these issues together with others specific to that domain, for example different languages and company culture.

ACKNOWLEDGMENTS

REFERENCES

[1] Ågerfalk, P., Fitzgerald, B., Holmström, H., Lings, B., Lundell, B. and Ó Conchúir, E. Framework for considering Opportunities and Threats in Distributed Software Development. In *Proceedings of the International Workshop on Distributed Software Engineering* (Paris, August 29, 2005). Austrian Computer Society, 2005, 47-61.

[2] Carmel, E. and Agarwal, R. Tactical approaches for alleviating distance in global software development, *IEEE Software, 18*, 2 (Mar./Apr., 2001), 22-29.

[3] Damian, D., Lanubile, F. and Oppenheimer, H.L. Addressing the Challenges of Software Industry Globalization: The Workshop on Global Software Development, In *Proceedings 25th International Conference on Software Engineering* (May, 2003). IEEE Computer Society, Los Alamitos, 793-794.

[4] Ebert, C. and De Neve, P. Surviving Global Software Development, *IEEE Software, 18*, 2 (Mar./Apr., 2001), 62-69.

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

[6] Herbsleb, J.D. and Moitra, D. Guest Editors' Introduction: Global Software Development. *IEEE Software*, *18*, 2 (Mar./Apr., 2001), 16-20.

[7] Herbsleb, J.D., Paulish, D.J. and Bass, M. Global Software Development at Siemens: Experience from Nine Projects, In *Proceedings of the 25th International Conference on Software Engineering – ICSE 2005* (May, 2005), IEEE Computer Society, Los Alamitos, 524-533.

[8] Jacobson, I. *Object-Oriented Software Engineering: A Use Case Driven Approach*. ACM Press, New York, NY, 1991.

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

[10] Lings, B., Lundell, B., Ågerfalk, P. and Fitzgerald, B. Ten Strategies for Successful Distributed Development, In *IFIP WG8.6 Working Conference on the Transfer and Diffusion of IT for Organisational Resilience*, Springer, (Galway, Ireland, June 7-10, 2006), *(to appear)*.

[11] Paasivaara, M. Communication Needs, Practices and Supporting Structures in Global Inter-Organizational Software Development Projects. In *International Workshop on Global Software Development* (May, 2003), pp. 59-63, <gsd2003.cs.uvic.ca/gsd2003proceedings.pdf>

[12] Paasivaara, M. and Lassenius, C. (2003) Collaboration Practices in Global Inter-organizational Software Development Projects, *Software Process Improvement and Practice, 8* (2003), 183-199.

[13] Prikladnicki, R., Audo, J., Evaristo, R. Requirements Management in Global Software Development: Preliminary Findings from a Case Study in SW-CMM context. In *International Workshop on Global Software Development* (May, 2003), 53-58, <gsd2003.cs.uvic.ca/gsd2003proceedings.pdf>

[14] Pyysiäinen, J. Building Trust in Global Inter-Organizational Software Development Projects: Problems and Practices, In *International Workshop on Global Software Development*, (May, 2003), 69-74, <gsd2003.cs.uvic.ca/gsd2003proceedings.pdf>

[15] Yin, R. *Case Study Research*. Sage Publications, Thousand Oaks, California, 2003.