Studying the Impact of Global Software Development Characteristics on Project Goals: A Causal Model

Ansgar Lamersdorf*[,1] and Jürgen Münch[,2]

[1]*University of Kaiserslautern, Software Engineering: Processes and Measurement Research Group, P.O. Box 3049, 67653 Kaiserslautern, Germany,*

[2]*Fraunhofer IESE, Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany*

Abstract: Despite the growing importance of global software development (GSD), a high failure rate of GSD projects is reported in the literature. In order to understand the underlying reasons, the goals of GSD projects and the special characteristics of GSD that might have an influence on these goals have to be identified. This article presents a combined literature and interview study aimed at identifying goals and influencing factors in GSD and integrates them into a causal model. The article presents the goal and design of the study; the literature review, which resulted in a preliminary model of factors and cause-effect relationships; and the revision of the model based on interviews with practitioners in GSD.

1. INTRODUCTION

In recent years, Global Software Development (GSD) has become more important, in both industry and research [1], [2], which is due to the benefits that a company distributing its work globally can achieve: global resource pools, attractive cost structures, the possibility of developing around the clock, and presence on local markets [2].

However, there is an increasing number of reports about problems in distributed projects caused by various characteristics inherent to remote and global collaboration [3-7], resulting in high failure rates of GSD projects [8], tendencies towards "nearshoring" [9], and suggestions not to start global development at all [10].

These findings indicate that there exist a number of factors that, on the one hand, have a significant impact on the success of GSD projects (i.e., the achievement of the project goals) and, on the other hand, are immanent to distributed development.

Detailed knowledge about these factors could help project managers to take them into account during project planning and thereby reduce the risks of distributed development projects. This applies especially to decisions about work organization and task allocation: If the factors that cause risks and problems in GSD projects is known and systematically taken into account in the decision on how to assign development work across globally distributed sites, the work can be distributed in a way that minimizes their negative impact [11]. For example, if a great cultural distance between sites is known to have a negative impact on productivity, then closely coupled work should not be assigned to sites with large cultural differences.

*Address correspondence to this author at the University of Kaiserslautern, Software Engineering: Processes and Measurement Research Group, P.O. Box 3049, 67653 Kaiserslautern, Germany;
E-mail: a_lamers@informatik.uni-kl.de

Thus, the question addressed in this article is: "Which goals exist for global software development projects and which factors impact these goals?" This question will be examined by a combined literature and interview study, which has resulted in a causal model that can be used as a basis for understanding cause-effect relationships as well as for further refinement, extension, or adaptation.

The remainder of this article is structured as follows: After presentation of the study goal and design in Section 2, the literature and interview study will be described in detail. The literature study results in a preliminary model, which will be revised based on the results of the interview study. The article concludes with a discussion of the results and its threats to validity.

2. STUDY GOAL AND DESIGN

2.1. Study Goal

The goal of the study is the development of a causal model of GSD project goals and characteristics of distributed development that have an impact on these goals. Accordingly, the main questions are:

Question 1: What are goals of GSD projects?

The aim of this question is to identify "typical" goals of distributed development projects and analyze if they differ from the goals of collocated projects.

Question 2: What are factors influencing the goals?

For this question, it is important to emphasize that we focus our attention on factors that are specific to distributed development: Factors considered either are only relevant if remote collaboration between multiple sites takes place (e.g., time shift between sites) or if they are necessary to describe the differences at the individual sites (e.g., process maturity at each site). This means that many factors that undoubtedly have an impact on project goals are not considered here since their impact is independent of the question of whether a pro-

1874-107X/10 2010 Bentham Open

ject is performed in a distributed manner or not: For example, the factors "Product complexity" and "Required reusability" are known to have an impact on project cost [12], which is, however, independent of the fact that a project is done in a collocated or distributed manner across sites A and B or across sites C, D, and E. They would thus not be regarded here.

Question 3: What is the impact of the influencing factors on the project goals?

Not all factors might influence all project goals; some influences might be positive while others are negative, and some factors might influence goals only indirectly via some intermediate factors. The aim of this question is to identify the cause-effect relationships and integrate them into one causal model.

2.2. Study Design

In order to improve the validity of the results, we decided to use a combined literature and interview study for gathering the results. First, we conducted a systematic literature review in order to get a preliminary answer to the research questions.

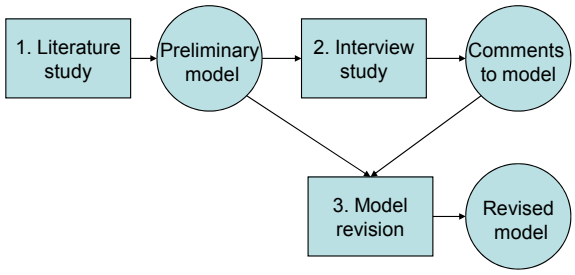

Fig. (1). Study process.

These results were then used as a baseline for the subsequent interview study: The findings of the literature analysis were presented to practitioners and experts in the field of distributed and global software development. Their answers and comments were then used for refining and adapting the model. This method of generating theory by applying two different types of studies and comparing the results is known in the literature as triangulation [13] and can help to make the theory more valid.

We chose the triangulation approach over a singular interview study or a systematic literature review for two reasons: On the one hand, we wanted to get a clear picture of the current state of the practice and thus decided to conduct interviews with practitioners instead of analyzing past experiences reported in the literature. On the other hand, since we had access to only a limited pool of interviewees, we aimed at extracting as much focused information as possible from the individual interviews. Therefore, we decided to use the preliminary results from the literature study for the interview study in order to focus the practitioners' answers.

Fig. (**1**) shows the overall study process. The detailed designs of the literature and interview study will be presented in their respective sections.

3. LITERATURE STUDY

In the following section, the interview study will be presented in detail, resulting in a preliminary model of goals and influencing factors.

3.1. Definition of the Study

The goal of the study was the development of a model that could be used as a baseline for the subsequent interview study. Thus, its main research questions are the same as in the overall study:

Question 1: What are goals of GSD projects?

Question 2: What are factors influencing the goals?

Question 3: What is the impact of the influencing factors on the project goals?

In order to answer these questions, published experiences from practical applications of GSD were analyzed by a systematic literature review. As a result, a list of goals, factors, and causal relationships was identified.

The experiences in the literature stem from various types of projects and focus on many different aspects. Thus, in order to reduce the complexity of the study results, abstraction had to be used: Information that only describes specific aspects of individual studies was not included in the results and similar phenomena were summarized under one notion (e.g., the factors "staff experience", "domain knowledge", and "design knowledge" will be summarized as "knowledge & expertise").

3.2. Literature Scope

3.2.1. Selection Process

A great amount of literature exists on the subject of distributed software development: GSD is discussed regularly in publications and conferences on software engineering, such as *Communications of the ACM* and the *International Conference on Software Engineering*. In addition, a growing number of special issues and conferences are dedicated to distributed development, such as two issues of *IEEE Software* and the *International Conference on Global Software Engineering*.

Thus, not all literature on global software development was considered. In order to qualify for the study, a publication needed to fulfill two requirements:

1) The results presented must be based on practical experiences. Ideally, this would be done by re-porting the experiences of case studies or regular projects in industrial practice. However, similar results can be achieved by conducting experiments or empirical studies. In some cases, results were included that seemed not to be based on direct practical experiences but on the opinions of experienced authors or on exhaustive literature studies.

2) The publication must describe problems and phenomena of global software development. Many papers describe tactics and technical solutions for reducing coordination problems in distributed collaboration. Even though these results can be very helpful in order to optimize GSD, they do not (or only indirectly) provide information on the factors

influencing GSD problems. Therefore, these papers were not considered.

The literature included in the study ranges from descriptions of industrial projects via controlled experiments to large-scale surveys. Depending on the origin of their results, three types of studies were distinguished:

Case studies: These studies describe the experiences of single distributed development projects. In most cases, the projects described were done in industry. However, some case studies were also done in an academic context, usually as student projects.

Empirical studies: Here, the aggregated results of many projects are reported. Mostly, this is done by analyzing data from a number of projects, by reporting the experiences of many project managers collected in interviews or surveys, or by conducting experiments focusing on individual aspects of distributed development.

Other: Some publications are not directly based on practical experience. As described above, these results usually should not be included in the literature study. However, single publications that report valuable results and seem to be based on the author's experiences or on a literature study were included.

Some papers report the experiences of several different case studies. Thus, it was hard to decide whether they belong to the category of case studies or empirical studies. In order to get a distinct classification, it was decided to classify all publications as empirical studies that report the experiences of more than three projects at once.

3.2.2. Included Data Sources

The literature included was gathered from different data sources. In all cases, the publications were selected by first looking for papers describing distributed or global software development. For every paper, the decision about whether it fulfills the requirements stated above was made based on its abstract. Sources of data were:

1) Web of Science by the Institute for Scientific Information; papers were searched by the keywords "Global Software Development" and "Distributed Software Development".

2) ProQuest Database; again the keywords for the search were "Global Software Development" and "Distributed Software Development".

3) Special issues of *IEEE Software* on global software development (volume 18, issue 2, March 2001 and volume 23, issue 5, September 2006).

4) Papers published at the International Workshop on Global Software Development for the Practitioner, located at the International Conference on Software Engineering.

5) Papers published at the International Conference on Global Software Development.

6) Papers referenced by already collected publications.

While many sources contained overlapping results (especially the two databases) and others did not contain any relevant results (as for instance the second special issue of *IEEE Software*), in the end 25 publications were included in the study (see Appendix for a complete list).

3.3. Results

The detailed list of publications and their relevant findings is given in the Appendix. In the following, the results will be presented as answers to the questions stated above.

3.3.1. Goals

The following goals were identified in the literature:

Costs: Minimization of the overall software development costs. Anticipated cost reductions are often described as one of the driving factors for outsourcing. In other cases, the goal is to deliver within budget [6, 14-20].

Time: Often, total development time reduction is one of the reasons for starting global software development. In these cases, projects try to establish follow-the-sun development. However, other publications describe problems in executing that model and report problems that increase the total development time [5, 17, 18, 20, 21].

Quality: As in other development projects, the quality of the product is also a goal in distributed development. Although quality usually is not directly affected by the distributed nature, some papers describe indirect effects of distributed collaboration on quality [7, 16, 20].

Resource utilization: Another important driver for GSD projects is the availability of proper human resources: Development work is given to remote sites because of a lack of available staff at the local site or in order to get the best people. This is summarized as resource utilization [15, 17-19, 22].

Proximity to customer: For many companies, success depends on close contact to customers. So proximity to customers often is a goal in distributed projects. A variation of this is reported as the goal of being globally present [15, 17, 23].

IP protection: Working with remote sites and external companies implies the risk of losing intellectual property (IP). Therefore, the goal of protecting IP can be of great importance in distributed development [14, 16].

3.3.2. Factors Influencing the Goals

The following relevant factors were identified:

Available resources: The number and capacity of the staff available at each site [17, 18, 22, 24].

Costs: The costs per head per hour at a certain site [6, 15-20]).

Physical distance: The distance between two involved development sites, affecting the impediments to traveling between sites [15, 20, 25].

Language differences: The language barrier between people at two different sites that occurs when the native language differs [6, 14, 15, 20, 24, 26].

Cultural difference: Differences in culture between people at sites located in different countries. Their impact has

been studied in different contexts [27] and exists also in software development [4, 6, 15, 20, 21, 24-26, 28, 29].

Organizational differences: Differences in organizational processes and culture if more than one organization is involved in distributed development [4, 21, 24].

Infrastructure distance: The infrastructure link between two sites (e.g., connection speed, connection reliability, shared tools) [4, 29].

Time shift: Time zone difference between sites [4, 21, 24].

IP security: The risk of losing intellectual property, e.g., by high staff turnover rates, cultural attitudes towards copying, or a lack of legal prosecution [14, 16].

Local government: Impediments or subsidiaries by the government or legal regulations at certain sites [15, 16].

Process maturity: The quality of the local development process at the involved sites, e.g., expressed by the CMM level [4, 16, 19].

History of working together: The common experiences of two sites, i.e., the amount of collaborative work in the past. [20, 25, 26, 30].

Knowledge at site: The amount of expertise and knowledge available at a site. Knowledge can be domain and/or technical knowledge [4, 6, 14, 18, 19, 24, 26, 31-33].

Coupling between tasks: For two tasks, this is the degree of collaboration needed between two teams that work on those tasks [5, 6, 34].

Needed knowledge for task: The knowledge (both technical and domain knowledge) that is needed for working on a certain task [20, 35].

These factors belong to different groups: Some factors describe the development sites; others describe the tasks that can be assigned to sites. Both the task and the site factors can be further classified into properties of tasks and sites and relationships between two tasks or sites.

Table **1** shows the resulting classification and the factors for each class.

Another set of factors was identified as influencing the goals but dependent on the other factors stated above:

Cost overhead: Overhead due to the distributed nature of the project, such as traveling costs between the sites [15, 18, 26].

Problems in communication, coordination, and control: A set of problems created by distributed development due to the impediments in communication [4, 14, 16, 20, 25, 26, 29, 31-34, 36].

Productivity: The amount of work done within a certain time [16, 30].

Lack of trust: Mistrust between different sites that causes further problems [6, 7, 14, 18].

Knowledge fit: The fit between the knowledge required for a certain task and the knowledge available at the site the task is assigned to [14, 19, 33, 35].

Table 1. Influencing Factors from Literature Study

Sites		Tasks	
Proper-ties	**Dependencies**	**Proper-Ties**	**Depen-dencies**
• Costs • Knowledge • Local govern-ment • Process matur-ity • IP security • Available resources	• Physical distance • Language difference • Cultural difference • Organizational dif-ference • Infrastructure dis-tance • Time shift • History of working together	• Needed know-ledge	• Coupling

3.3.3. Relationship between Factors and Goals

The following relationships were identified:

Physical distance → cost overhead: Higher distance increases overhead for traveling [15].

Cost overhead → costs: Higher cost overhead leads to higher total development costs [21].

Physical distance → proximity to customer: Distances between sites have a negative impact on the connection of the remote sites to the customer [14, 21].

Time shift → total development time: An increased time shift between sites makes round-the-clock development possible. This may decrease total development time [21, 36].

Time shift → problems in communication, coordination, and control: Time zone differences decrease the opportunities for synchronous communication and thereby increase the problems [16, 33, 36].

IP security → IP protection: The success of protecting intellectual property depends on the IP security at the involved sites [14, 16].

Available resources → resource utilization: Optimal utilization of resources obviously depends on the resources available at the sites [17, 18, 22].

Local government → cost overhead: Legal regulations and local taxes depend on the local government at the sites and impact the cost overhead [15, 16].

Development process → quality: Poor development processes can cause low product quality [16].

Proximity to customer → quality: If the distance to the customer is great, the quality of the requirements may suffer, which in turn decreases product quality [16].

Productivity → costs: High productivity decreases development costs for a product and vice versa [16].

History of working together → problems in communication, coordination, and control: If teams know each other

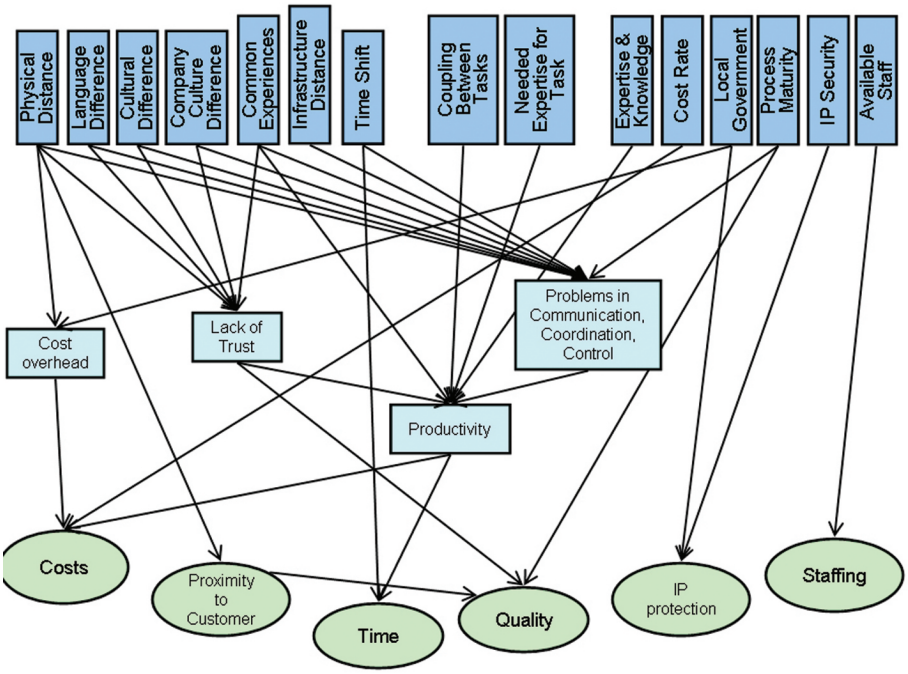

Fig. (2). Causal model as a result of the literature study.

due to prior experiences, they can communicate more efficiently and with fewer problems [20, 25, 30].

Cultural difference → problems in communication, coordination, and control: Problems can be caused by cultural misunderstandings [4, 25, 26, 32].

Organizational difference → problems in communication, coordination, and control: Differences in organizational processes and cultures impede communication [4, 25].

Physical distance → problems in communication, coordination, and control: Distance between sites decreases the possibilities of communication [31].

Language difference → problems in communication, coordination, and control: If teams do not communicate in their native language, problems in understanding can occur [26, 33].

Infrastructure distance → problems in communication, coordination, and control: Good connection speed between sites increases communication efficiency [29, 33].

Process maturity → problems in communication, coordination, and control: Low process maturity can increase the problems; high maturity can help to overcome them [4].

Problems in communication, coordination, and control → productivity: Productivity suffers from increased problems [25, 30, 34].

Knowledge at site → knowledge fit: The fit of required and available knowledge obviously depends on the available knowledge [14, 19, 33].

Needed knowledge → knowledge fit: Likewise, the knowledge fit depends on the knowledge needed for a task [35].

Knowledge fit → productivity: If not all required knowledge is available at a site, it has to be learned, which decreases productivity [14, 19, 33, 35].

Coupling between tasks → productivity: High coupling between tasks located at different sites means much communication is needed between sites and thereby decreases efficiency [34].

Language differences → lack of trust: Misunderstandings can cause irritations that de-crease the trust [7].

Cultural differences → lack of trust: Trust can also suffer from cultural aversions or misinterpretations [7].

Organizational differences → lack of trust: Misunderstandings due to different organizational cultures and processes can increase lack of trust [7].

Physical distance → lack of trust: Separation between sites decreases the chance of informal contacts and can increase aversion between teams, which increases mistrust [6].

History of working together → lack of trust: Common experiences and established relationships can establish trust between sites [21].

Lack of trust → productivity: Lack of trust between sites hinders efficient collaboration and thereby decreases productivity [7, 20].

Lack of trust → quality: Product quality may also suffer from a lack of trust [7, 20].

Productivity → total development time: High productivity decreases the time needed for development [16].

3.3.4. Overall Model

The factors and goals and the relationships between them create a first causal model as shown in Fig. (**2**).

4. INTERVIEW STUDY

The results of the literature study were used as input to a subsequent interview study. This study will be described in the following section.

4.1. Definition of the Study

The goals of the interview study were the same as for the literature study: In order to develop a causal model, goals for GSD projects, influencing factors, and their relationships had to be identified.

However, as a preliminary causal model already existed, the research questions were not to identify goals, factors, and relationships from scratch, but to revise the existing results from the perspective of experienced practitioners.

Thus, the specific research questions were:

What are the experts' comments to the baseline model?

Which goals, factors, or relationships would they add? Which would they remove?

Out of every set of factors influencing other factors and goals, which ones are seen as most and least important?

Based on the answers of the experts, the aim was to build a revised causal model based on the findings from both literature and practice.

4.2. Research Methodology and Participants

4.2.1. Research Methodology

Interviews were selected as the main research methodology. Depending on the communication between the partners, interviews can be divided into telephone interviews and personal interviews with both persons facing each other [37]. Telephone interviews have several advantages over personal interviews, mainly due to the ease of access: Since the interviewer does not have to travel to the interviewee, the costs of telephone interviews are much lower. Additionally, a large number of interviews can be conducted within a short amount of time. It is also easier to access interviewees who are reluctant to sacrifice time for a personal meeting.

On the other hand, there are many disadvantages of telephone interviews compared to personal ones: Since only spoken words are transmitted over the telephone, it is harder to get information. This reduces both the possibility of the interviewer to formulate complex questions and his or her ability to obtain all of the interviewees' responses (e.g., their mimic action). The respondent might also not be willing to discuss sensitive topics over the telephone. Besides, the interviewer has less control compared to a personal interview. The interviewee can, for example, easily terminate the interview by simply hanging up the phone.

Thus, it is suggested [37] to use the telephone interview as an alternative only in certain circumstances. We tried to follow this guideline by conducting as many interviews as possible in person and only used telephone interviews if there was no other option. We also tried to minimize the problem by mailing the questionnaire to the interviewees in advance so that they had all questions in front of them. This helped very much to avoid misunderstandings during the interviews.

During the interview, we used a detailed questionnaire that contained both open and closed questions. The questions aimed at first identifying the background of the interviewee (e.g., his or her experience and the company's history in GSD) and then obtaining his or her comments on the baseline model. This was done by presenting the model as a whole and afterwards focusing specifically on every causal relationship (e.g., "We found the following factors influencing productivity: […]") and asking (a) about the relative importance of the factors (by ordering them) and (b) if the interviewee would remove some factors or add other factors to this relationship.

All answers were recorded and, with the exception of one, transcribed literally. For the remaining one, detailed notes were taken. The interview transcription and the notes were then analyzed using coding: Pieces of the answers were categorized and grouped together in order to identify commonalities and differences. This was done using the NVivo software [38].

4.2.2. Study Participants

In total, 10 subjects were interviewed for the study. The interviewees came from 9 different companies, mostly based in the US. Most of the participants had experiences in distributed software development in middle or senior management positions (e.g., project, quality, or product manager). Other positions included chief architect or process analyst. The interviewed persons had many years of experience in distributed development – the majority reported at least five years of experience, and two of them had been involved in distributed development for nearly 20 years.

Participants were selected by contacting personal contacts of the authors. In addition, practitioners attending the International Conference on Global Software Engineering 2008 were asked to contribute to the study.

4.3. Results

4.3.1. Goals for GSD Projects

All interviewees confirmed that the six goals presented (costs, time, quality, staffing, intellectual property protection, proximity to customer) were, in principle, valid and reasonable, and they did not mention any further goals. However, even though all practitioners agreed on proximity to customer being a benefit to the project, the interviews showed that proximity was not a goal per se but rather a constraint or a criterion used for achieving better productivity or quality.

Intellectual property protection is also a goal that might be questionable as part of a model: Even though all interviewees agreed to this goal in principle, it did not seem relevant to most of them. Only a small number reported that IP protection was one of their concerns but then refused to give further details on that.

The "typical goals" of software development projects – costs, time, and quality – were agreed upon by all interview-

ees, but the relative importance of the goals seemed to be different de-pending on the context.

Staffing as a special goal of distributed development seemed to be of high importance in distributed development, which matches the experiences reported in the literature. However, similar to proximity to customer, staffing seemed to be more a constraint for task assignment than a project goal.

4.3.2. Development Costs

When asked on the relative importance of cost rate, productivity, and cost overhead on the development costs, all interviewees agreed on productivity having an important impact. The importance of the cost rate seemed to depend on whether the work was being distributed globally or only within the US. For those who worked only with sites in the US, the cost rate was not relevant, since all sites had equal cost structures. People doing distributed work with Asia, on the other hand, judged the cost rate as a very important factor for the development costs.

Cost overhead was described to the practitioners as costs for traveling and for local overhead structures at the sites, such as additional taxes. (Overhead due to increased communication and decreased productivity was not included here.) With that in mind, most people said that the cost overhead was negligible. Only one interviewee reported a significant traveling overhead (4-5% of the budget), but in all other cases traveling costs were not important.

4.3.3. Development Time

The literature study showed that the total development time of a GSD project would, on the one hand, suffer from a productivity decrease due to distributed collaboration. On the other hand, a time shift between sites could be used for "round-the-clock" development, in which different sites with mutually exclusive shifts would work on one item and thereby reduce the development time.

Confronted with these two influences on the development time, all practitioners agreed that there is a development time increase due to decreased productivity.

"If you are on an entirely different working schedule and the working hours don't overlap, then you are given emails that cannot be resolved until the next day so you almost have a day delay just trying to do the communication."

The positive influence of a time shift on development time (by using it for round-the-clock development) was seen much differently. While some people saw it as not really being useful, others reported successful applications of round-the-clock development that helped to reduce development time. Apparently, the success of development around the clock is dependent on other factors: Only certain combinations of tasks (e.g., coding and testing of one component) allow for applying the necessary daily handovers. In addition, many practitioners reported that very high process maturity is needed to implement efficient development around the clock.

In general, many of the interviewees seemed to doubt the possibility of significantly reducing development time by round-the-clock work.

4.3.4. Quality

Three influence factors for the quality of the development were given to the interviewees: process maturity, lack of trust, and proximity to customer.

Another factor that came up in many of the interviews was the impact of expertise and knowledge on quality. Most of the practitioners believed that this influenced the quality more than any of the other three factors:

"With very talented people you make up for all sorts of problems."

Different terms were used for that factor – talent, expertise, training, and others. But overall, there were only two types of expertise meant by these terms: general software development capability and expertise and specialized technical or domain knowledge for a specific task.

Out of the three given factors, most of the practitioners weighted process maturity and lack of trust as having the biggest impact on quality, with slightly more people seeing process maturity as most important. However, the term "lack of trust" was not interpreted equally by all people: Some saw lack of trust as a personal relationship between people of different sites which influenced quality negatively; others interpreted it as uncertainty about the ability of the remote group. Therefore it is hard to evaluate the answers concerning lack of trust as a factor.

One interesting answer came from one manager of an Indian company. Judging the biggest influence factors for quality, he answered:

"The answer to this question differs on which part of the world you are. If you are in the Anglo-Saxon world, it is process maturity. If you are in the orient, it is proximity to customer and lack of trust."

Proximity to customer was seen as having less influence on quality by nearly all interviewees. One of them stated that the impact of proximity on quality would depend on the type of the task:

"The proximity to customer is really important when you are in the initial phases where you are trying to understand what they need, what they want. So I would say this depends on the phase of the development project and what work you are trying to do."

4.3.5. Productivity

As to influences on productivity, four different factors were given to the interviewees: knowledge fit, lack of trust between sites, problems in communication between sites, and coupling between tasks assigned to different sites.

Out of these four, problems in communication and knowledge fit were evaluated as being most important.

Regarding the relative weight between knowledge fit and communication problems, the practitioners were split into two groups. Some of them saw the communication problems as being most important and often weighted even the coupling between tasks higher than knowledge fit – one of them said about knowledge problems:

"This one I think you can overcome. […] The reason is, sometimes you can have processes that can help you."

However, the larger group weighted the expertise and knowledge available at the site as having the highest impact on productivity.

Coupling between tasks was seen as having a slightly smaller impact on productivity.

Only one person named lack of trust as having a big influence on productivity. For all of the others, it was a minor factor and some reported that it did not have any impact at all.

4.3.6. Lack of Trust

Five factors influencing lack of trust were given to the practitioners: physical distance, language differences, cultural differences, differences in company culture, and common experiences. Probably because of a different understanding of the term "lack of trust", the weights given to these factors differed widely. However, the common experiences or the history of working together between sites was judged by most as having the biggest impact.

Differences were eminent in the weights for the factors of language, cultural, and company culture distances. To some, cultural and language differences (with more emphasis on the cultural differences) were the factors causing most problems and mistrust. One manager, for example, who worked with a group of Chinese, complained about the Chinese culture, which, from in his point of view, led to misunderstandings and mistrust in their work results:

"There is a tendency in the Chinese culture that if they don't understand something, they don't tend to ask. […] And as a result, there have been several issues where people have spent days doing work and if they had asked a question, they could have done it in hours."

Other practitioners, however, did not see major problems due to language or culture.

Differences in company culture were considered to have an important influence for some people. Sometimes this was traced back to differences in the national culture of the companies. To others, it did not have a major impact.

Nearly all people judged physical distance as not having a big impact.

Overall, lack of trust seems to be very hard to capture in a model because it is hard to define, weighted very differently, and caused by many different factors. In addition, it is not easy to describe trust on the level of sites, since in many cases, it was built from personal relationships between individuals.

4.3.7. Problems in Communication, Coordination, and Control

The factors identified in the literature study as having an impact on communication problems were physical distance, language differences, cultural differences, differences in company culture, common experiences of working together, infrastructure distance, time shift, and process maturity.

Out of these factors, time shift was mentioned most often as having the biggest impact on communication problems. Time shift between sites caused many problems because of delayed communication:

"People sometimes send you [a] request for information and they need immediate response to be able to do their job that day. When they have to wait a whole day, it is a killer."

Process maturity was also weighted as having a big influence. Apparently, with mature processes on both sides it is much easier and more efficient to exchange information on the status of work. Although usually not weighted as having the highest impact, infrastructure was mentioned as an important influencing factor by nearly all of the interviewees. This shows that the presence or absence of proper communication tools and network links can very much influence the efficiency of communication.

The factors of language and culture were weighted differently, depending on the context of the interviewee: Practitioners who had to work within large globally distributed environments saw them as big impediments to efficient communication, while others who did not experience them much did not weight them as important. This seems to indicate a tendency of underestimating problems with language and cultural differences.

Common experiences of working together had a major impact to some persons. For one interviewee, it was even the biggest of all influencing factors. However, others did not weight it very highly. Company culture differences were mostly seen as not important or as part of the cultural differences.

Nearly all judged the physical distance between sites as the least important influencing factor for communication problems.

4.3.8. Revision of the Literature Study Model

Based on the results of the interview study, several changes were applied to the previous causal model:

1) The literature model included the goal of "Proximity to customer". However, the interviews showed that proximity to customer is rather a characteristic of a task: Some tasks have to be near the customer in order to work productively and with quality, while for other tasks, it does not matter if they are done near the customer. Therefore, proximity to customer was removed as a goal and introduced as a characteristic of a task.

2) "IP protection" was another goal of the model. Most of the interviewees agreed with this, but all of them said that for their company, it was not important. As the goal thus does not have much importance and no further information about it could be captured from the interviews, it was removed.

3) "Staffing" was also included as a goal. While optimal resource utilization was confirmed by most of the practitioners as a major company goal, for single projects it seems to be more a constraint than a goal: Tasks can only be assigned to sites where re-sources are available. It was therefore removed as a goal.

4) Cost overhead due to physical distance (for traveling costs and local regulations) was mostly judged as negligible

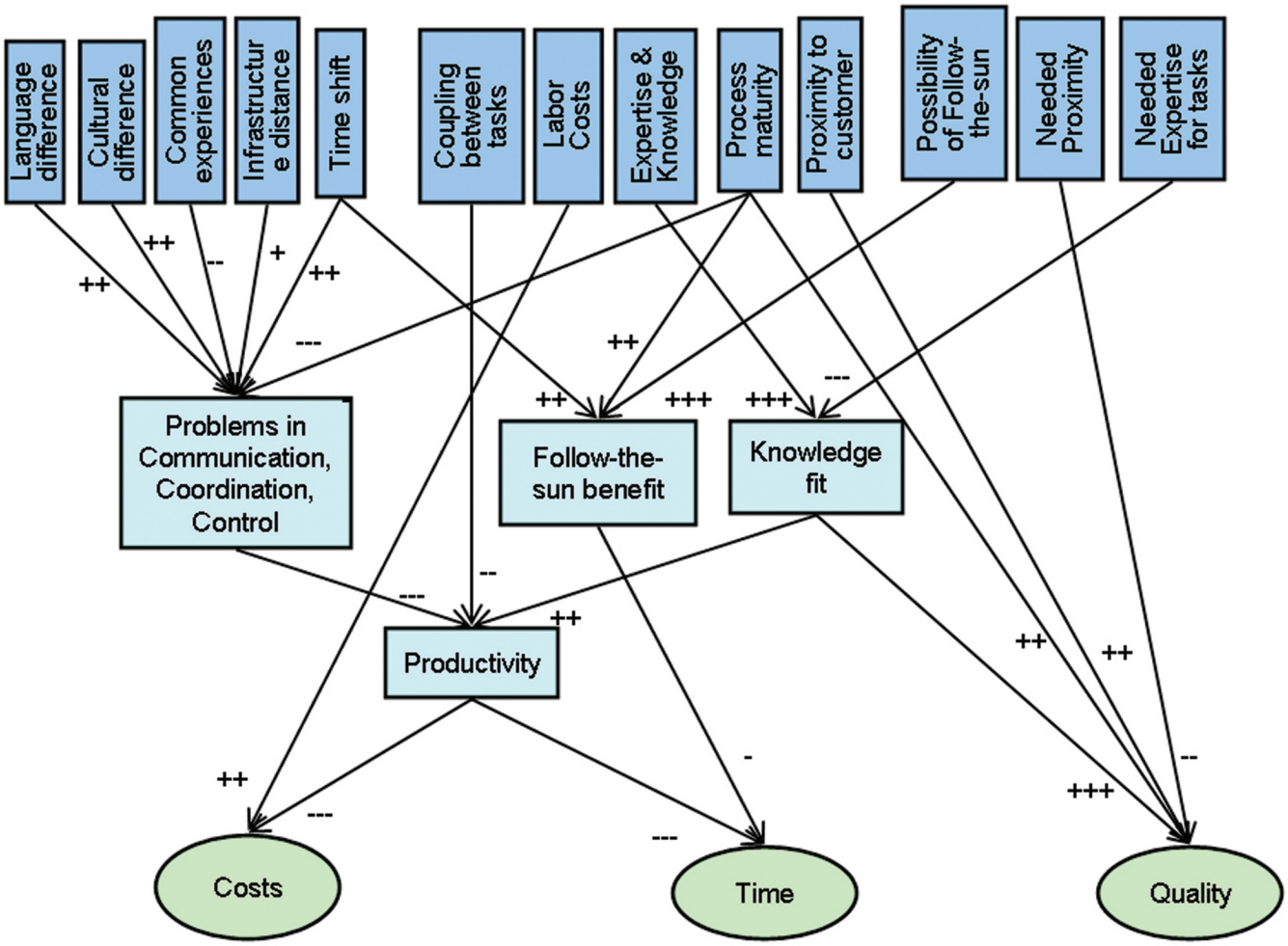

Fig. (3). The revised model after the interview study. Relationships are weighted with low, medium, and high (+, ++, +++) positive or negative (+, -) impact.

and therefore removed. The factor "local government" was thus also removed.

5) "Lack of trust" was really hard to model. As shown above, the term was interpreted differently by the interviewees and had many similarities to the interpretation of communication problems, so it was removed from the model. In future work, the term will probably have to be defined more specifically and investigated further.

6) "Physical distance" was seen as negligible by most of the practitioners and thus removed.

7) "Company cultural differences" was seen as unimportant by many persons. Others saw it as very similar to "cultural differences". It was thus decided to summarize both factors under "cultural differences".

8) The interviews showed that a time shift did not automatically have a positive impact on development time due to round-the-clock development. Instead, a benefit from round-the-clock development can only be achieved if the task allows it and a certain process maturity exists. The model was extended to capture this.

9) Nearly all of the practitioners reported that quality was strongly influenced by the knowledge and expertise of the local workforce. Thus, a relationship between "Expertise & knowledge" and "Quality" was added.

10) Relative weights were given to the causal relations according to the experts' rankings. If the majority of the experts judged a relation as being more important than the others, it was weighted with +++ (---, respectively), if it was seen as less important by the majority, it was weighted with + (-), and ++ (--) was given otherwise. However, as the rankings given by the experts varied to a large extend, the results have to be handled with care.

The revised model is shown in Fig. (**3**).

5. DISCUSSION

The summary of the identified results is already given in Fig. (**3**). The following sections will discuss these results by naming threats to validity, drawing conclusions, and identifying open research questions.

5.1. Threats to Validity

Even though the results are grounded in both a systematic literature review and a subsequent interview study, there is a set of threats to validity that have to be considered when

interpreting the data. These will be discussed in the following.

Most of the papers analyzed in the literature study have a special focus on specific topics within GSD. Therefore, the probably of disregarding factors and relationships was relatively high. For instance, very obvious relationships like the impact of productivity on costs were described in only one paper. Other relationships were not mentioned in any of the publications even though they very likely do exist. (e.g., process maturity having an impact on productivity). In part, this threat was weakened by the revision of the model in the interview study. However, as the already existing model was given to the interviewees, it is likely that they often simply agreed to the given factors and relations instead of coming up with their own detailed experiences.

The number of interviewed practitioners is relatively low. In addition, they came from very different organizations and backgrounds. This raises the question of whether the results can be generalized to all kinds of global software development. We tried to keep the described factors and phenomena on a relatively high level, but still it is likely that in concrete environments, other factors might be more relevant.

This is also supported by the fact that many relationships (e.g., the impact of cultural differences on communication problems) were seen quite differently by the practitioners: While some saw the impact of one specific factor as very high, others found it almost negligible. Thus, it is clear that a detailed casual model can only be done within one specific environment.

The possibility of misinterpretations represents another threat to validity: It was not possible to keep the factor "lack of trust" within the model because it was understood differently by the practitioners. However, several studies [6, 7, 14, 18] report that it does have a significant impact on project success. Similarly, people might have had a different understanding in interpreting other factors and delimiting them (e.g., the factors "language difference", "cultural difference", and "company culture difference").

Another threat is grounded in the fact that the interviews were given to an outsider. This might have prevented the practitioners from reporting all problems openly. For example, some refused to talk about problems with intellectual property protection, which resulted in the removal of this factor even though it might have a significant influence in the practice of GSD projects.

5.2. Conclusion and Open Research Questions

The goal of this study was to develop a causal model in order to identify the characteristics of global software development that make it different from collocated development and that lead to its large number of problems and project failures.

Fig. (**3**) shows the resulting model. Even though, as shown before, the weight of the factors is judged differently by the practitioners and a detailed model would probably look different in every organization, it gives a good overview of the factors that cause problems in GSD.

The model contains many already well-known factors and influences, as for example the impact of process maturity on quality. We, however, believe that it still can provide important information for researchers and practitioners since it, on the one hand, gives insight into the impact of well-known factors on GSD-specific problems and benefits (e.g., the influence of process maturity on the possibility of follow-the-sun development) and, on the other hand, makes explicit that an organization has to consider various amounts of factors (and not just labor cost rates [41]) while deciding on starting GSD and on the organization of distributed development.

The impact of all of the factors identified here does not only depend on whether a project is globally distributed or not, but also on the question of what the distribution of work across sites looks like: For example, the impact of task coupling and cultural differences on productivity is higher in a scenario in which closely coupled tasks are assigned to sites with large cultural differences compared to a scenario in which only tasks with loose coupling are assigned to different sites that have only low cultural differences between them.

This demonstrates that a distribution of work across sites (i.e., a task assignment), which systematically takes into account all the factors identified here (including their impact on project goals), can help to reduce the problems of global distribution and thereby improve the success of GSD projects.

However, we discovered that in practice, work assignment is done rather unsystematically and takes into account only few factors, such as the availability of the workforce and local cost rates [39]. This discrepancy between decision criteria and the fact that, at the same time, the practitioners reported many problems caused by the factors identified here indicates a potential for improvement.

We thus see a need for systematic decision support in GSD task allocation that can help practitioners to efficiently take into account all relevant influencing factors for their allocation decision and thus decrease the immanent risks of global software development. In doing so, models like the one developed here are needed for understanding the impact of allocation decision on project success. As of now, we have already developed a decision model and an implementation in the TAMRI (Task Allocation based on Multiple cRIteria) tool that can help to identify work distribution for specific scenarios [11, 40]. The causal model identified here was used as the baseline for this work. In the future, we plan to extend the decision support models to include other relevant influences on task allocation such as characteristics of the developed products.

In order to provide decision support for specific environments, detailed causal models and influencing factors have to be developed that reflect the characteristics of GSD within these specific organizations. We thus see the work presented here only as a starting point for our research on the characteristics on global software development and on ways to improve the success rates of GSD projects by providing systematic decision support in GSD project initiation and task assignment.

ACKNOWLEDGMENTS

The authors would like to thank all participants of the interview study for providing insights into the practices of distributed software development. Professor Carolyn Seaman, Professor Dieter Rombach, and Professor Peter Freeman provided valuable help and comments. Most of the work was done during a stay at the Fraunhofer Center for Experimental Software Engineering, Maryland and was financially supported by the Otto A. Wipprecht Foundation. The authors also thank Sonnhild Namingha for proofreading the paper.

APPENDIX: LITERATURE INCLUDED IN SURVEY

List of References for Each Study Type

Case Studies	Ebert & De Neve (2001) [15], Casey & Richardson (2006) [6], Treinen & Miller-Frost (2006) [21], Battin *et al.* (2001) [22], Lindqvist *et al.* (2006) [31], Heeks *et al.* (2001) [32], Kobitzsch *et al.* (2001) [33], Mullick *et al.* (2006) [35]
Empirical Studies	Alami *et al.* (2008) [14], Gareiss (2002) [18], Kommeren & Parviainan (2007) [19], DeLone *et al.* (2005) [20], Pilatti *et al.* (2006) [5], Smite & Moe (2007) [7], Ramasubbu & Balan (2007) [23], Smite (2004) [24], Oza & Hall (2007) [26], Komi-Sirvio & Tihinen (2005) [29], Herbsleb & Paulish (2005) [4], Espinosa *et al.* (2007) [30], Herbsleb & Mockus (2003) [34], Espinosa *et al.* (2007) [36]
Other	Sakthivel (2007) [16], Carmel (1997) [17], Gurung & Prater (2006) [25]

REFERENCES

[1] J. D. Herbsleb and D. Moitra, "Guest editors' introduction: Global software development," *IEEE Softw.*, vol. 18, no. 2, pp. 16-20, Mar./Apr. 2001.

[2] D. Damian and D. Moitra, "Global software development: How Far Have We Come?" *IEEE Softw.*, vol. 23, no. 5, pp. 17-19, Sept./Oct. 2006.

[3] E. Carmel and R. Agarwal, "Tactical approaches for alleviating distance in global software development," *IEEE Softw.*, vol. 18, no. 2, pp. 22-29, Mar./Apr. 2001.

[4] J. D. Herbsleb, D. J. Paulish, and M. Bass, "Global software development at Siemens: Experience from nine projects," In: *International Conference on Software Engineering (ICSE)*, St. Louis: USA, pp. 524-533, 2005.

[5] L. Pilatti, J. Audy, and R. Prikladnicki, "Software Configuration Management over a Global Software Development Environment: Lessons Learned from a Case Study," In: *International Workshop on Global Software Development for the Practitioner*, Shanghai: China, 2006.

[6] V. Casey and I. Richardson, "Uncovering the reality within virtual software teams," In: *International Workshop on Global Software Development for the Practitioner*, Shanghai: China, 2006.

[7] D. Smite, and N. B. Moe, "Understanding a lack of trust in global software teams: A multiple-case study," In: *International Conference on Product Focused Software Development and Process Improvement PROFES*, Riga: Latvia, pp. 20-34, 2007.

[8] M. Fabriek, M. Brand, S. Brinkkemper, F. Harmsen, and R. W. Helms, "Reasons for success and failure in offshore software development projects," In: *European Conference on Information Systems*, Galway: Ireland, pp. 446-457, 2008.

[9] E. Carmel and P. Abbott, "Why 'nearshore' means that distance matters," *Commun. ACM*, vol. 50, no. 10, pp. 40-46, October 2007.

[10] G. Seshagiri, "GSD: Not a business necessity, but a march of folly," *IEEE Softw.*, vol. 23, no. 5, pp. 63-64, Sept./Oct. 2006.

[11] A. Lamersdorf, J. Münch, and H. D. Rombach, "A decision model for supporting task allocation processes in global software development," In: *International Conference on Product Focused Software Development and Process Improvement PROFES*, Oulu: Finland pp. 332-346, 2009.

[12] B. Boehm, C. Abts, A. Brown, S. Chulani, B. Clark, E. Horowitz, R. Madachy, D. Reifer, and B. Steece, *Software Cost Estimation with COCOMO II*. Prentice-Hall, 2000.

[13] C. B. Seaman, "Qualitative Methods," in *Guide to Advanced Empirical Software Engineering*, F. Shull, Ed. Springer, 2008.

[14] A. Alami, B. Wong, and T. McBride, "Relationship issues in global software development enterprises," *J. Glob. Inf. Technol. Manage.*, vol. 11, no. 1, pp. 49-68, 2008.

[15] C. Ebert and P. De Neve, "Surviving global software development," *IEEE Softw.*, vol. 18, no. 2, pp. 62-69, Mar./Apr. 2001.

[16] S. Sakthivel, "Managing Risks in Offshore Systems Development," *Commun. ACM*, vol. 50, no. 4, pp. 69-75 April 2007.

[17] E. Carmel, "The explosion of global software teams," *Computerworld*, vol. 31, no. 49, C6, Dec. 8, 1997.

[18] R. Gareiss, "Analyzing the Outsourcers," *Information Week*, Nov. 18, 2002.

[19] R. Kommeren, P. Parviainan, "Philips experiences in global distributed software development," *Empir. Softw. Eng.*, vol. 12, no. 6, pp. 647-660, Dec. 2007.

[20] W. DeLone, J. A. Espinosa, G. Lee, and E. Carmel, "Bridging global boundaries for IS project success," In: *38th Hawaii International Conference on System Sciences*, p. 48.2, 2005.

[21] J. J. Treinen, S. L. Miller-Frost, "Following the sun: Case studies in global software development," *IBM Syst. J.*, vol. 45, no. 4, pp. 773-783, Oct-Dec 2006.

[22] R. D. Battin, R. Crocker, J. Kreidler, and K. Subramanian, "Leveraging resources in Global Software Development," *IEEE Softw.*, vol. 18, no. 2, pp. 70-77, Mar./Apr. 2001.

[23] N. Ramasubbu and R. K. Balan, "Globally distributed software development project performance: an empirical analysis," In: *6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering*, Dubrovnik: Croatia, pp-125-134, 2007.

[24] D. Smite, "Global software development project management – distance overcoming," In: *European Conference on Software Process Improvement (EuroSPI)*, Trondheim: Norway, pp.23-33, 2004.

[25] A. Gurung and E. Prater, "A Research Framework for the Impact of Cultural Differences on IT Outsourcing," *J. Glob. Inf. Technol. Manage.*, vol. 9, no. 1, pp. 24-43, January 2006.

[26] N. V. Oza and T. Hall, "Difficulties in managing offshore software outsourcing relationships: an empirical analysis of 18 high maturity indian software companies," *J. Inf. Technol. Case Appl. Res.*, vol. 5, no. 3, 2005.

[27] G. Hofstede, *Culture's Consequences – Comparing Values, Behaviors, Institutions and Organizations across Nations*. 2nd Edition, Thousand Oaks, London, New Delhi, 2001.

[28] E. McGregor, Y. Hsieh, P. Kruchten, "Cultural Patterns in Software Process Mishaps: Incidents in Global Projects," In: *Workshop Human and Social Factors of Software Engineering (HSSE)*, St. Louis: USA, pp. 1-5, 2005.

[29] S. Komi-Sirvio and M. Tihinen, "Lessons learned by participants of distributed software development," *Knowl. Process Manage.*, vol. 12 no. 2, pp. 108-122, 2005.

[30] A. Espinosa, S. A. Slaughter, R. E. Kraut, and J. D. Herbsleb, "Familiarity, complexity, and team performance in geographically distributed software development," *Organization Science*, vol. 18, no. 4, pp. 613-630, Jul./Aug. 2007.

[31] E. Lindqvist, B. Lundell, and B. Lings, "Distributed development in an intra-national, intra-organizational context: an experience report," In: *International Workshop on Global Software Development for the Practitioner*, Shanghai: China, 2006.

[32] R. Heeks, S. Krishna, B. Nicholson, and S. Sahay, "Synching or sinking: global software outsourcing relationships," *IEEE Softw.*, vol. 18, no. 2, pp. 54-60, Mar./Apr. 2001.

[33] W. Kobitzsch, H. D. Rombach, and R. L. Feldmann, "Outsourcing in India," *IEEE Softw.*, vol. 18, no. 2, pp. 78-86, Mar./Apr. 2001.

[34] J. D. Herbsleb and A. Mockus, "An empirical study of speed and communication in globally-distributed software development," *IEEE Trans. Softw. Eng.*, vol. 29, no. 6, pp. 481-494, June 2003.

[35] N. Mullick, M. Bass, Z. Houda, D. J. Paulish, M. Cataldo, J. D. Herbsleb, and L. Bass, "Siemens global studio project: experiences adopting an integrated GSD infrastructure," In: *International Conference on Global Software Engineering*, Florianopolis: Brazil, pp. 203-212, 2006.

[36] J. A. Espinosa, N. Nan, and E. Carmel, "Do gradations of time zone separation make a difference in performance? A first laboratory study," In: *International Conference on Global Software Engineering*, Munich: Germany, pp. 12-22, 2007.

[37] D. Nachmias and C. Nachmias, *Research Methods in the Social Sciences*. St Martin's Press: New York, 1987.

[38] QSR International: *NVivo 8 – Research Software for Analysis and Insight*. Available at: http://www.qsrinternational.com/products_nvivo.aspx [Accessed Oct. 15, 2008].

[39] A. Lamersdorf, J. Münch, and H. D. Rombach, "A survey on the state of the practice in distributed software development: criteria for task allocation," In: *International Conference on Global Software Engineering*, Limerick: Ireland, pp. 41-50, 2009.

[40] A. Lamersdorf, J. Münch, "TAMRI: a tool for supporting task distribution in global software development projects," In: *International Workshop on Tool Support Development and Management in Distributed Software Projects*, Limerick: Ireland, pp. 322-327, 2009.

[41] M. Bass, D. Paulish, "Global software development process research at siemens," In: *Third International Workshop on Global Software Development*, Edinburgh: Scotland, UK, pp. 8-11, 2004.

Received: September 25, 2009 Revised: October 15, 2009 Accepted: May 03, 2010