

Exploring the Assumed Benefits of Global Software Development

Eoin Ó Conchúir, Helena Holmström, Pär J. Ågerfalk, Brian Fitzgerald
*Lero – The Irish Software Engineering Research Centre,
University of Limerick, Limerick, Ireland.
{eoin.oconchuir, helena.holmstrom, par.agerfalk, bf}@ul.ie*

Abstract

In existing global software development (GSD) literature, much focus has been on identifying the challenges that practitioners may face (such as socio-cultural and temporal distance issues), while potential benefits have not been extensively analyzed. We reverse this trend by studying these potential benefits. We question whether they are well-founded assumptions and whether they are attainable in practice. This paper presents findings from a multi-case study at three multi-national companies that have extensive experience in GSD. We identify the benefits mentioned in GSD literature, analyze them with regards to the companies' experiences and then conclude whether or not each benefit is being realized in practice. Our findings reveal that the realization of the assumed benefits cannot be simply taken for granted.

1. Introduction

Global Software Development (GSD) is becoming increasingly common practice in the software industry [1]. There are many potential benefits from GSD, including reduced development costs due to the salary savings possible [2, 3]; reduced cycle time arising from 'follow-the-sun' software development [4, 5] and time-zone effectiveness [6]; new opportunities for cross-site modularization of development work [7]; access to a larger and better-skilled developer pool [3, 8, 9]; innovation and shared best practice [6]; and closer proximity to markets and customers [9, 10]. However, GSD also introduces a number of challenges in relation to communication, coordination and control of the development process. These arise due to the distances involved in three major dimensions – geographical, temporal, and socio-cultural. As a consequence, much research and practice has focused on trying to find ways to overcome these GSD challenges. Typically, potential benefits of GSD are only mentioned and their realization taken for granted. Here we reverse this trend: we merely list the main challenges of GSD and

focus instead on the benefits and the extent to which they are actually being realized in practice in three global companies practicing GSD.

2. Challenges of Global Software Development

GSD is characterized by stakeholders from different national and organizational cultures, located in separate geographic locations and time-zones, using information and communication technologies to collaborate. Not surprisingly, such conditions introduce major challenges in relation to team communication, coordination and control [11]. As can be seen in previous studies, temporal, geographical and socio-cultural distances give rise to a number of GSD challenges, for example:

- Temporal distance reduces overlapping hours of possible collaboration, and can introduce a delay in feedback from colleagues [12]. The use of asynchronous communication tools may not adequately handle ambiguities [13], and can increase the risk of misunderstandings [14].
- Geographical distance reduces informal communication [4]. This can hinder the development of a sense of 'teamness', the establishment of trust and the spread of important information about the project.
- Cultural background can greatly affect how developers interpret and respond to a certain situation [11]. Language difficulties often introduce misunderstandings [8] thus hampering effective communication and coordination in GSD.

3. Research Question and Methodology

Our study focused on the potential benefits of GSD as identified in peer reviewed literature [11], namely reduced development costs, increased time-zone effectiveness, cross-site modularization of development work, access to a large skilled labor pool, and innovation and shared best practice. We aimed to explore the extent to which industry practitioners are realizing these benefits, if at all. We were motivated

by the reasoning that the true benefits of GSD must be properly understood if we wish to fully realize them.

A multi-case study [15] was performed in three multi-national companies, which are described below. Data collection involved twelve semi-structured qualitative interviews, through which we aimed to learn from the experiences of those actively involved in the practice of GSD. The interviews were of one hour duration, with follow-up email contact used to refine issues as they emerged. Those interviewed included site managers, project managers, a project architect, team leads, software engineers and technical support staff - all interviewees were directly involved in GSD activities at the companies. A primarily qualitative grounded theory (GT) approach was adopted for data analysis of the transcribed interviews [16].

Complementary to the interviews, on-site meetings were held as well as university-hosted workshops. See Table 1 below for a summary of research activities. After our first round of interviews in January-March 2005, preliminary findings were presented to the companies at a university-hosted workshop. This was followed by supplementary interviews in July-August 2005, allowing for more in-depth exploration of the research topic. Preliminary findings were reported in [17].

Table 1. Summary of research activities.

Date	Research Activity
Jan 2005-Mar 2005	On-site meetings and interviews at company sites
Jun 2005	Workshop at University
Jul 2005-Aug 2005	Interviews at company sites

4. Case Setting

Our study focused on three companies actively practicing GSD – Intel, Fidelity Investments and HP. Each company has its headquarters in the US with software development teams in Ireland, and all directly involved in intensive GSD. The interviews reported in this study were conducted at the Irish sites of these companies. Further studies of this nature would benefit from interviews at all sites involved in any GSD project, across continents. The three companies were chosen because of their established GSD activities, each company having a software development site within driving distance of the university campus.

The Irish Intel site we studied employ 125 people and is part of Intel’s Infrastructure Processor Division, with GSD teams based at several international sites, including the US, Malaysia, China, India, and Poland.

Fidelity Investments provides financial services and investment resources internationally and is one of the largest private companies in the US. The company has been developing software at this site in Ireland since 2001, and currently employs around 100 people there. The software products developed are supplied to internal customers in the US, by coordinating with several software development teams in the US and India. Hewlett-Packard (HP) provides desktop support services right through to mission critical service delivery. The Irish team develops remote support and proactive services, as part of a 130 person R&D and software development site. The team is closely integrated with colleagues located in the US, effectively creating a virtual team across multiple sites. Software development work is also being carried out in India.

5. The Assumed Benefits of GSD

Here we identify the six main benefits that have been associated with GSD in peer-reviewed literature. Based on the multi-case study findings, we present an analysis of each benefit and conclude on whether or not practitioners can expect to realize these benefits to their full potential.

5.1. Reduced Development Costs

One of the most important reasons for organizations to embark on a challenging and risky endeavor such as GSD is, not surprisingly, its potential to reduce development costs [2, 3, 6]. By moving parts of the development work to low-wage countries, the same work can be done for a fraction of the cost.

In our study, all three companies stated that reducing costs was one of the main drivers for GSD. Within Fidelity, the Indian internal customer billed half of what the Irish team charged for essentially the same work. A base annual salary of US\$15,000 for a software developer in India, is one quarter of the salary of an Irish developer, who in turn earns half that of a developer in the US. At Intel, the same scale of savings between the US, Ireland and India was applicable. The HP site manager reported a ten-times saving when offering remote customer support rather than sending an engineer to the customer’s site. A further three-times savings could be achieved when doing this remote support work from a low-wage economy. However, a Fidelity manager warned that team objectives and planning must be very precise before going offshore, as otherwise it only adds to complexities and problems that may already exist in

the co-located team, and hence offsetting the original expected costs benefits.

While the companies were at different stages of offshoring, they all tended to send non-complex development work to low-cost regions. As one manager said,

“Basic development should be done in India because it is lower value add, and lower value-add work belongs where it can be done more cheaply.”

Another manager also said that the “onshore” workers could send less interesting work, like fix enhancements, to India, so as to be able to work on more complex and interesting work, but the manager also warned,

“You have to give the Indians good work, since the turnover rate is very high, and they could get more interesting work elsewhere. They want pure development work.”

The site manager at HP stated that in order for GSD to become more efficient, they might need to give the Indian site responsibility for complete projects instead of sections of projects.

An eight-fold salary saving (as in the case of US versus India above) seems like a significant up-front benefit. However, coordination costs increase when developers are distributed. As one HP manager warned, *“developers may be cheaper off-shore, but we must look at productivity.”* The manager bemoaned the lack of models for calculating the true cost of distributing development. A manager in Fidelity accepted that it makes good sense to go offshore from a cost savings point of view, however,

“If there’s a struggle between communications or requirements in the existing team, adding the India component to it only compounds the issue. It makes it worse. We need to be very clear on what we’re doing and have our processes locked down.”

Furthermore, GSD seems to create demand for an increased number of managerial roles. Indeed, an HP manager said that their India operation had a very top-heavy management structure. At this company – where they strive for a ‘virtual team’ environment – a manager may be responsible for the activities of developers located remotely. This requires an additional on-site “host manager” at the remote site. The Fidelity team was also employing this managerial structure. The increased demand for employing on-site host managers, may erode the benefit of distributing software development globally in order to access lower-paid developers.

The drive to save costs in these companies also affected the amount of travel between sites. This meant that most developers had never met their remote colleagues. With no face-to-face communication, it may be very difficult to create a feeling of teamness

and to establish trust. Interestingly, however, one senior manager at Intel stated, *“It’s essential that developers travel.”* When asked whether developers meet each other, a manager at HP said, *“Travel restrictions would never allow that.”* Therefore, the companies were saving money when employing cheaper developers offshore, but at the same time, these developers did not get the chance to meet each other, thus constraining the possibility of building effective long-term relationships with remote colleagues.

Also, as a result of having remote colleagues on lower wages, there can be perceived threat that they will eventually replace developers in the higher-cost economies [4]. The managers in our study did try to overcome this fear. A manager in Fidelity described how he briefed his team,

“[GSD is] just a fact of life. But what I had to say to them in taking on the team in India was there is a future for us [in Ireland], there’s a future for India. We have to work together to work out what it is. It doesn’t have to be a win-lose situation.”

Companies taking on GSD should be aware that even if cost savings will be achieved, it may take some time to get the full development activities up and running. The managers in Fidelity noted that they needed to build up the “critical mass” of the team in India, where there was limited experience in the particular project. In taking on new developers located in India, one manager said that they would not partake in any GSD effort for their first three months.

Hence, our study shows that reduced development costs are indeed a driver for GSD. However, although cost savings can be achieved in terms of lower average wage per man-hour, there are other aspects that need to be taken into consideration. For example, our study reveals the lack of models to calculate true costs, additional managerial overhead, perceived threat from lower paid colleagues, and additional time to build up a ‘critical mass’. Also, restrictions on travel between sites in order to limit costs can lead to problems such as the lack of trust, and possibly, reduced productivity. Companies practicing GSD may not be able to easily achieve cost savings as drastically as pure salary comparisons may suggest.

At the same time, it is undeniable that cost savings are achievable, a fact that is shown by the massive trend of sourcing software development labor globally. Cost-benefit tradeoffs for GSD are still not well understood [18]. Further research needs to establish return on investment (ROI) figures for globally-distributed software development projects.

5.2. Leveraging Time-Zone Effectiveness

Having developers located in different time-zones can allow organizations to increase the number of working hours during a day, hence decreasing cycle time. This is known as follow-the-sun development, and is often assumed a potential benefit of GSD [4, 5, 10]. Even in development projects that do not operate for 24 hours of the day, having developers working at one site while developers of another site sleep is attractive, at least in principal [18].

Even though the companies in our study certainly acknowledged the possible benefits of time-zone effectiveness, such a scenario was seen as difficult to realize. Achieving time-zone effectiveness implies that developers are located across several time-zones with, ideally, some overlapping work-hours during the day. However, sufficient over-lap in working hours may be difficult to achieve due to, for example, lunch breaks or national holidays. Hence, delays in response may prolong the development process. A manager in HP said that there could be frustration when awaiting a reply from other sites. During critical phases of the software lifecycle, an HP engineer stated, *“If you’re trying to progress quickly, it [time-zone difference] is a problem.”* Indeed, in Intel, the managers tried to limit that extent of temporal distance:

“We distribute teams across a maximum of two geos [geographies], which makes time-zones more manageable.”

According to the companies, only limited types of activities may be suitable for the follow-the-sun model. As one Intel manager said,

“Follow-the-sun is not practical for software development. It might be more practical to distribute different phases, like testing.”

Another manager in Fidelity agreed,

“Follow-the-sun may be not suited for development work, but is good for defect resolution and support.”

Interestingly, the companies seemed to strive for a model diametrically opposed to follow-the-sun development – they tend to shift their working hours in order to maximize the number of overlapping work-hours across sites. Indeed, it seems that the companies view time-zone differences not as a potential benefit but as a negative side effect of GSD. The HP team reduced temporal distance by effectively creating a virtual day across their virtual team. A manager in HP commented about their flexible work practices:

“People go out of their way to work late at night. I regularly have calls with US workers at 6am, and I also work quite late. The official workday [in Ireland] is 8.30am to 5.15pm, but that’s not applied at all. Taking calls at home can become quite intrusive on

one’s family and personal life. In the long run, you get burnout of people.”

In Fidelity, they also managed to increase the number of overlapping work hours. Given a 5 hour time difference between the US and Irish sites, with US developers beginning at 6am local time and Irish workers beginning at 9am local time, time-zone difference was effectively reduced to only two hours.

Despite being a widely assumed benefit, time-zone effectiveness was not a benefit being realized by the companies in our study. As mentioned by our respondents, delayed responses and the fact that all development phases are not suitable for ‘follow-the-sun’ development make this hard to achieve, and is ultimately an unattractive approach to GSD. Instead, they prefer to focus on the overlapping work hours available to them, and to shift work hours to increase the number of overlapping hours across sites. It has been previously suggested that global teams need to be flexible in their work hours, and to “share the pain” with some teams working late and others working early [19]. This can, in turn, affect the personal lives of the developers, and possibly lead to a burnout of people.

5.3. Cross-Site Modularization of Development Work

The nature of GSD forces teams to split their work into well-defined individual modules, which is beneficial to the overall development as these can be developed in parallel [6, 7]. At the same time, teams need to be wary of reduced communication between sites, leading to problems at the integration stage [20]. In earlier work by Parnas we have seen the importance of a separation of concerns when decomposing work into modules [21]. Conway noted that the structure of the system reflects the structure of the organization that developed it [22]. It appears that these principles could be extremely relevant again in the specific case of GSD.

We found varying approaches with regard to the modularization of work. The HP team practiced the ‘virtual team’ model, treating all team members as members of one large team while being physically separated by great distance. In Intel, the approach was to explicitly modularize tasks by feature. They treated one set of collocated colleagues as one team, with all teams coordinating to achieve the completion of the end product. One manager in Intel explained,

“We try to have as few dependencies as possible on other teams’ work. In general, we try to have feature dependency orthogonal across sites.”

By adopting a ‘modularizing by feature’ approach, Intel could also capitalize on phasing the release of software as it is completed:

“We build blocks of software in each site, and then the blocks fit together in the end. Individual releases can be made available to the customer without every block being prepared. The customer can choose which components to use [as the development work progresses].”

Indeed, Intel also decided not to over-modularize work by limiting the distribution of teams within a project to a maximum of two global areas (such as Europe and Asia, for example). They recognized that co-location of team members is needed to develop certain units of functionality.

Fidelity also noted advantages with modularizing development work, especially compared with developing by the follow-the-sun method, as it seems less complex to achieve. Discrete ‘chunks’ of work can be sent to the remote site, providing that site with some level of ownership within the project, thus improving the sense of goodwill. However, the degree of modularity of work must be considered, ensuring that the tasks are large enough to offer increased efficiency when modularized. One Fidelity manager said,

“In one of our projects, we tried to distribute every task. But some things were too small to distribute. There was no efficiency.”

Furthermore, modularization of work can create integration problems. For example, if remote teams become too independent, with a lack of inter-site communication during the development stage, there may be difficulty in integrating their work in the end when incorrect assumptions about functional requirements come to light. As one member of the technical staff at Intel warned,

“You don’t want your team to become too isolated, especially in a small team such as we have here in Ireland.”

A software engineer in Intel recalled a large drop in communication once their project was distributed:

“Much less inter-site communication happened because of distribution. We had formal weekly project status meetings, and then used phone conversations during overlapping work hours.”

A general sense of cross-site teamness amongst developers did not seem to emerge:

“Because of the split of work according to features, we remain generally two different teams.”

Overall, it should be noted that while the modularization of work due to geographical distribution of developers can be effective in reducing the required level of cross-site communications, it might also be an obstacle to the creation of a sense of cross-site teamness. Also, our study shows that

modularization of work is not always preferable. For example, there is no efficiency gained in modularizing smaller tasks. Companies must carefully assess the advantages and disadvantages of various models of allocating tasks across sites [9].

5.4. Access to Large Skilled Labor Pool

GSD has the potential to facilitate access to a large pool of highly skilled workers [2, 3, 8-10]. Accordingly, all three companies highlighted the fact that GSD allows them to access a larger labor pool, with specialized skills, particularly in countries such as India and Malaysia. A manager in Fidelity noted,

“India is considered a good place to have the Quality Assurance (QA) skill set. The testing component is typically being done in India.”

A manager in Intel added that the scalability available to them as a result of access to a large labor pool allows them to increase greatly the size of their development efforts without dramatic changes to the organization. The manager stated,

“It’s very difficult to hire skilled engineers [locally]. When you don’t have the capacity to complete some specific work, outsourcing offers us scalability.”

The site manager in Intel was particularly pleased with the high skills that graduates obtain in India and Malaysia. In India, Intel can recruit relatively low-waged graduates from the top four or five universities in India, choosing Ph.D. graduates, resulting in access to “genius employees.”

However, the benefit of having access to a large low-cost and skilled labor pool may not be automatically realized. Due to rapid growth in the employment market for software developers in Bangalore, for example, the companies reported problems of very high attrition levels.

Other disadvantages linked to seeking out workers in other employment markets are due to the implied increase in cultural distances between team members. As cultural differences are increased among team members, existing problems may be exacerbated. All three companies noted various problems due to cultural differences within their GSD teams. For example, one of the site managers we interviewed had issues with dealing with his staff in India:

“Language difference is a really big problem. Half the time, I don’t know what the guys are saying, quite honestly.”

An architect at one of the sites offered another example of problems due to different cultural backgrounds:

Table 2. Extent of Realization of GSD Benefits

Assumed Benefit	Extent to which Realized	Overall Verdict
Reduced development costs	Eight-fold developer salary differential between US and India for example. Taking advantage of lower-cost sites for less complex and non mission-critical tasks. Significant overhead in communication, coordination and control.	Partially realized
Leveraging time-zone effectiveness	Time-zones not a benefit but a cause of reduced collaborative time window and unusual working hours. Follow-the-sun not used for development activities, but sometimes for other activities, such as testing.	Not realized
Cross-site modularization of development work	Modularization of work by features across sites can be effective in reducing the required level of cross-site communications. Might also be an obstacle to the creation of a sense of cross-site teamness.	Partially realized
Access to large skilled labor pool	GSD does provide access to large pool of skilled workers. Extremely high attrition levels in rapid growth regions. Not all the desired skill-sets may be readily available. Socio-cultural problems abound.	Partially realized
Innovation and shared best practice	Employees who feel threatened by low-wage colleagues are unlikely to share more than necessary to get the job done. Cross-site standardized processes may help.	Partially realized
Closer proximity to market and customer	Although local presence provides for better access to customers, cultural problems internally increase accordingly.	Partially realized

“I was trying to get people to use the same tools. I said, ‘This is like a religious war.’ One of the guys in the US said, ‘Well, I’m highly religious, I go to church,

with customer-interfacing employees in the US, while their actual development was still taking place in India.

GSD does indeed offer unprecedented possibilities to companies to expand their software development activities, and to quickly form virtual global organizations [8]. Despite being assumed as a clear-cut benefit, access to a large labor pool does have some drawbacks that should not be ignored. In our study, respondents report on challenges such as high attrition levels due to rapid growth in Asian economies and hence, high turn-over in staff. Also, an increased cultural distance brings with it problems difficult to foresee.

5.5. Innovation and Shared Best Practice

It has been suggested that due to GSD actors coming from various backgrounds, organizations can take advantage of increased innovation and shared best practice amongst team members [6]. A manager in Fidelity suggested,

I really care about it, so what do you mean?’ They took me up completely wrong.”

The Intel site managed to reduce visible cultural distance at times, by hiring outsourcing companies

“Having people coming from different backgrounds will always help, getting different views from different people, since people coming from different parts of the world would have different ways of doing something.”

A software engineer at Intel also acknowledged the effect of working with people from different cultural backgrounds:

“When working a lot with people in another country, I even found my accent or thought process changing!”

However, in reality, it didn’t seem like GSD developers in our study ever had much opportunity to share best practices with each other. Lack of face-to-face contact inhibited informal communication, and reduced sharing of ideas between different sites. While managers had met many of their counterparts at other sites, many developers had never met their remote colleagues. This was the case at Intel, and even at HP where the team was striving for a ‘virtual team’ environment where geographical distance should not act as a boundary. With the lack of free flow of information between sites, it would be difficult for innovation and shared best practices to emerge.

Finally, some of the employees in the companies recognized the fact that the abilities of developers in lower-waged locations were possibly underestimated, an issue that has been previously highlighted [9, 19]. With a lack of respect for others' abilities, it is less likely for them to learn best practices from others.

Even though innovation and shared best practice is assumed a benefit of GSD, we found that this is seldom realized. Despite recognizing the value of working with people from different cultural backgrounds, the working environment does not always allow for interaction and exchange of ideas. Also, an underestimation of other peoples' skills may prohibit the true potential of shared best practice. It should be noted that elsewhere it has been reported that best practices may spread from the offshore vendor back to the home organization, as a result of increased focus on improving their Capability Maturity Model (CMM) levels to match that of the vendor's [23].

5.6. Closer Proximity to Market and Customer

By establishing subsidiaries in countries and on continents where one's customers are located, a more direct interaction becomes possible [8-10]. However, only one of the companies in our study noted the benefit of locating development efforts closer to their target market. As Intel is mainly a manufacturing company, many of their technology customers are located in China. Having local employees located in China, they are close culturally and linguistically to the customer, and have better knowledge of local business conditions. However, having developers located in the customers' market implies that there will be a cultural divide amongst team members – which would introduce the socio-cultural problems discussed above. Therefore, companies locating some of their development efforts in local markets in order to be closer to their customers must also develop strategies for overcoming socio-cultural problems.

6. Conclusions

While there are many significant beneficial aspects of GSD, our study clearly shows that these benefits are neither clear-cut nor can their realization be as taken-for-granted as the GSD literature may lead one to believe. Specifically, anyone engaging in GSD should be aware of the many risks associated with these benefits. Do not assume that overall costs will be reduced as much as salary savings suggest, as lower wages are countered with the overhead of higher managerial complexities. Pure follow-the-sun software development seems unrealistic, and the companies

prefer to modularize work instead of trying to take advantage of developers being situated in various time-zones. Seeking out employees in rapid growth markets can backfire, with very high attrition rates reported. Sharing of best practice between cultures can be problematic, especially if the ones sharing feel they are giving away their competitive edge to lower waged colleagues. Closer proximity to foreign markets leads to closer proximity to socio-cultural problems. Table 2 summarizes the main insights gained from this study in terms of how the assumed benefits of GSD played out in the studied organizations.

7. Acknowledgements

This research project has been financially supported by the EU FP6 project, Calibre and SFI principal investigation projects, B4-STEP and Lero.

8. References

- [1] United Nations, "World Investment Report 2004 - The Shift Towards Services," New York and Geneva 2004.
- [2] E. Carmel and R. Agarwal, "Tactical Approaches for Alleviating Distance in Global Software Development," *IEEE Software*, Vol. 18, No. 2, 2001, pp. 22-29.
- [3] D. Damian, F. Lanubile, and H. L. Oppenheimer, "Addressing the Challenges of Software Industry Globalization: The Workshop on Global Software Development," *Proc. 25th International Conference on Software Engineering*, Portland, Oregon, 2003.
- [4] J. D. Herbsleb and R. E. Grinter, "Splitting the Organization and Integrating the Code: Conway's Law Revisited," *Proc. 21st International Conference on Software Engineering*, Los Angeles, California, United States, 1999.
- [5] E. Carmel, *Global Software Teams: Collaborating Across Borders and Time Zones*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall, 1999.
- [6] C. Ebert and P. De Neve, "Surviving Global Software Development," *IEEE Software*, Vol. 18, No. 2, 2001, pp. 62 - 69.
- [7] M. Bass and D. Paulish, "Global Software Development Process Research at Siemens," in *International Workshop on Global Software Development*. Edinburgh, Scotland, 2004.
- [8] J. D. Herbsleb and D. Moitra, "Guest Editors' Introduction: Global Software Development," *IEEE Software*, Vol. 18, No. 2, 2001, pp. 16-20.
- [9] R. E. Grinter, J. D. Herbsleb, and D. E. Perry, "The Geography of Coordination: Dealing with Distance in R&D

Work," *Proc. International Conference on Supporting Group Work*, 1999.

[10] J. D. Herbsleb, A. Mockus, T. A. Finholt, and R. E. Grinter, "Distance, dependencies, and delay in a global collaboration," *Proc. 2000 ACM conference on Computer supported cooperative work*, Philadelphia, Pennsylvania, United States, 2000.

[11] P. J. Ågerfalk, B. Fitzgerald, H. Holmström, B. Lings, B. Lundell, and E. Ó Conchúir, "A framework for considering opportunities and threats in distributed software development," *Proc. International Workshop on Distributed Software Development*, Paris, France: Austrian Computer Society, 2005, pp. 47-61.

[12] J. D. Herbsleb, D. J. Paulish, and M. Bass, "Global Software Development at Siemens: Experience from Nine Projects," *Proc. 27th International Conference on Software Engineering 2005*, St. Louis, Missouri, USA., 2005.

[13] D. Damian and D. Zowghi, "The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization," *Proc. IEEE Joint International Conference on Requirements Engineering*, Los Alamitos, 2002.

[14] L. Kiel, "Experiences in Distributed Development: A Case Study," *Proc. ICSE International Workshop on Global Software Development*, Portland, Oregon, USA, 2003.

[15] R. K. Yin, *Case Study Research: Design and Methods*, 2nd ed. California, USA: Sage Publications, 1994.

[16] J. Corbin and A. Strauss, *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. California: Sage, 1990.

[17] E. Ó Conchúir, H. Holmström, P. J. Ågerfalk, B. Fitzgerald. "Global Software Development: Never Mind the Problems – Are There Really Any Benefits?" *Proc. 29th Information Systems Research Seminar in Scandinavia*, 2006.

[18] J. A. Espinosa and E. Carmel, "The Effect of Time Separation on Coordination Costs in Global Software Teams: A Dyad Model," *Proc. 37th Hawaiian International Conference on System Sciences*, Big Island, Hawaii, 2004.

[19] R. D. Battin, R. Crocker, J. Kreidler, and K. Subramanian, "Leveraging Resources in Global Software Development," *IEEE Software*, No. March/April 2001, 2001, pp. 70-77.

[20] R. E. Grinter, "Recomposition: putting it all back together again," *Proc. ACM conference on Computer supported cooperative work*, Seattle, Washington, United States, 1998.

[21] D. L. Parnas, "On the criteria to be used in decomposing systems into modules," *Communications of the ACM*, Vol. 15, No. 12, 1972, pp. 1053 - 1058.

[22] M. E. Conway, "How Do Committees Invent?," *Datamation*, Vol. 14, No. 4, 1968, pp. 28-31.

[23] J. W. Rottman and M. C. Lacity, "Twenty Practices for Offshore Sourcing," *MIS Quarterly Executive*, Vol. 3, No. 3, 2004, pp. 117-130.