Quality Issues in Global Software Development

Sanjay Misra
Dept. of Computer Engineering, Atilim University
Ankara, Turkey
smisra@atilim.edu.tr

Luis Fernández-Sanz
Dept. of Computer Sciences, Universidad de Alcalá
Alcalá de Henares, Spain
luis.fernandezs@uah.es

***Abstract*— The most advantageous features of Global Software Development (GSD) are its cost saving benefits and the easily availability of resources. Also the technological advancement especially in Information and Communication Technology (ICT) makes GSD a common practice in software industry. But GSD is also facing a lot of challenges. Maintaining quality in software development processes and products in GSD environments is one of the major challenges. This paper presents a survey on the challenges and factors which impact on the quality of the products in GSD environments. This report identifies that most of the factors which affect the quality of software product appear as part of two major challenges: requirements and coordination. We further demonstrate that how these two challenges are affected by several factors. Finally, we present the possible solution to reduce the complexity of those various factors.**

Keywords-Global Software Development; process quality; product quality; requirement and coordination challenges.

I. INTRODUCTION

The last decade is the evidence for the changing trends from the traditional software developments process, which was mainly confined on in-house software development, to the Global Software Development (GSD), where whole development process is distributed at different locations all around the world. In fact, advancement in tools and techniques in software development process has allowed geographically and culturally diverse groups to come together in global software development teams [1]. Further, the technological improvement in Information and Communication Technology (ICT) helps GSD to become a common practice in software industry. More specifically, Internet changed the whole society set up and; also make GSD as one of the most popular trends amongst software community.

There are several benefits of GSD. The most important ones which attract to the companies are cost and easy access to human resources. Although these are major advantages, several challenges also exist in GSD which makes difficult maximizing the benefits. One of the major challenges is how can one control the quality, while assigning the task at different locations where the actual work on project or sub projects are going to be performed. In fact, a unique feature of software product is that different parts cannot be developed as isolated activities. This is because there are complex dependencies between several tasks/parts of project; therefore development team members communicate with each other to fulfill their tasks during the whole development process [2]. In case of GSD, tasks are carried out at different locations distributed all around world, so communication suffers [3], and, as a result, the quality of product is also affected.

One can find several papers which address the different types of challenges in distributed software development such as contextual, cultural, organizational, geographical, temporal, and political [4]. However, the particular work on quality issues involved in GSD has not been researched yet to the appropriate extent. There are only few papers which addresses quality issues specifically. Ivček et al. [4] have presented a paper 'Aspects of Quality Assurance in Global Software Development Organization' in which the authors reported a case study of a project developed in two offshore countries. In their work they explained the way for applying quality assurance techniques in GSD environments. In another paper [5], the authors have explored whether working with others in GSD environment really matters for the quality of the software that is produced in global settings.

The lack of proper works for quality issues in GSD environments motivate us to work in this area. In this paper, we focus more particularly into quality issues: especially how quality of the final product is affected by the different factors involved in GSD. The observations reflect that, in general, the most of the challenges and issues of GSD seriously affect the quality of product. This paper presents a systematic study on how these challenges affect the quality and the ways to cope with these challenges. This paper is primarily a survey report which tries to identify the quality issues and their possible solutions in a GSD environment.

The paper is organized as follows. In the following sections, we start our investigation of quality challenges in software process and product in a GSD environment. In the same section, we also discuss and identify the several issues which affect process and product qualities. The solutions for the common problems which are responsible for quality are given in Section 3. The discussion on the work and the conclusion are given in Section 4 and Section 5, respectively.

II. QUALITY OF SOFTWARE PRODUCT IN A GSD ENVIRONMENT

IEEE defines the software quality as the degree to which a system, components or processes meet the specified requirements [6]. A modified form of this definition is: the degree to which a system, component or process meets customer or user needs or expectation [7]. In other words, quality means conformance to fulfilling the customers' requirements [8]. Juran [9] defined quality as "Quality

 ISBN: 978-1-61208-165-6

consists of those product features which meet the needs of customers" and thereby provide product specification. Further, Pressman [10] defines the software quality as conformance to explicitly stated functional and performance requirements, explicitly documented development standards and implicit characteristics that are expected of all professional developed software. If we combine and analyze all these definitions, we can observe that quality is mainly concerned with conformance to requirements. In general, the quality in requirements is achieved by using proper and effective elicitation techniques, which can provide the correct requirements.

The quality in product in a GSD environment is also affected by the type of software process as well as the existence of proper control of different activities within the software development process. However these, above activities i.e., collecting requirements and proper control of software process activities are not easy tasks in GSD environments. In globally distributed environment, several factors such as intercultural factors language barriers (i.e., the communication problems) reduce the chances of collecting the proper and sufficient requirements and make more difficult the coordination amongst the distributed team. As a result, it becomes difficult to control the quality in the processes at different locations of the world. Bartelt et al. [2] also support our claim that the key challenge of global software engineering is to establish appropriate communication and coordination habits in a global project environment [11]. However, in our opinion, communication is the main for proper coordination.

If we analyze how to achieve the quality objectives in global software distributed environment, firstly we have to evaluate how we can achieve the quality in eliciting the correct requirements. This, which is really not an easy job in distributed environment where customers and developers might be physically far away to each others. Secondly, we have to control quality in the software development process.

In summary, we can conclude that:

1. Quality in software product in GSD environment is challenged by the two major factors (Table 1).
   a. The lack of sufficient and correct requirements (due to several factors of GSD) and
   b. The lack of proper coordination in software development process.

TABLE I. QUALITY AND CHALLENGES IN GSD

	Quality in Product in GSD Environment	
	Requirement Quality	Processes Quality
Identified Challenges	Requirement [12] [13]	Coordination [11] [2]

The conclusion of above discussion is that the collection of proper requirements and coordination are main challenges in achieving quality in software development process in GSD environment. There are several factors which are responsible for these challenges. We will discuss these challenges and factors in more detail in the next section.

III. CHALLENGES FOR QUALITY IN GSD: THE REQUIREMENT AND COORDINATION AND THE FACTORS EFFECTING THEM

In the previous section, we have concluded that quality in GSD environment is challenged by requirement-related issues and problems in coordination. There are other factors, which impact the quality of the process and product in GSD environment. We treat them as factors of these two main challenges.

Gathering requirements from customers is always a challenge for software engineers. The quality of the product is highly affected by the absence of clear requirements. Guzman et al. [14] stated that normally there is a lack of common understanding of goals and requirements amongst development team members in GSD environments. Further, the growing number and the volatility of requirements also create additional challenges. Lormans et al. [12] stated that the evolution of requirements is usual: no matter how thorough the requirements specification has been set up, the requirements for any non-trivial system will change not only after the system has been built but also during the process of implementing the system. Further, the paper highlighted that the evolution of requirements appears due to a great variety of reasons [13]. In fact, requirements change and evolve mainly due to changing the requirement of business needs and also due to change in technology. Requirements are also modified in the process of designing, implementing, and writing test cases for requirements. These all examples show that there is always a need of modification of the initial set of requirements throughout the lifecycle of the project: and, if customers are far away from the development site, it becomes a challenge.

The agile software development approach appears in practice as a way to overcome the above problem. However, the principles of GSD are contradictory with the philosophy of agile development: in GSD, teams are distributed all around the world and customers are not necessarily be at the same site of development while agile promotes development, where the software is developed through iterative process with close involvement of the customers representatives. Agile approach is a widely accepted practice in software development and it claims getting more successful process in comparison to other practices. One of the reasons for this success rate is that customers and the development team members' work together more affectively. This technique reduces the understanding of customer's requirements up to maximum level. In case of any ambiguity in requirements they can solve the problems immediately and therefore reduce the time to take decisions [15]. In an opposite situation, the customers and development team members in GSD work in distant places from each other. This distance creates lots of problems in development, e.g., communication, language, time zone, etc. In the case of changes, modifications, or clarification of any requirement, it is not very easy to have an agile and effective action. Therefore, one cannot pretend the goal of agility in GSD except if the agile development processes are used separately at the nodes of the software development team.

 ISBN: 978-1-61208-165-6

International Standard Organization (ISO) also correlates software quality with the requirements and defines the software quality assurance activity [16] as a set of activities designed to evaluate the processes by which the software products are developed in order to satisfy stated or implicit requirements. IEEE [17] also focuses in satisfying requirements for Software Quality Assurance (SQA). It provides uniform and minimum acceptable requirements for preparation and contents of SQA plans [18]. These standards emphasize that satisfying requirements is the prime objective in quality assurance. On the other hand, the general problems involved in GSD (e.g., languages, cultural and time zone differences and lack of proper knowledge management) are responsible of arising difficulties in requirement engineering tasks. Further, the physical distance between stakeholders and group members limit their face-to-face interaction and thus create special challenges in the communication. As a consequence, this directly reduces the quality of requirements and therefore the quality of the product [19].

Coordination and control of the software development sites, distributed all around the world is also a major challenge in GSD. Quality of the development process is highly affected by the lack of proper coordination amongst the team members. There are several issues, which make this job as one of the most challenging in GSD. For example, ambiguities in understanding the organizational processes, the management practices, the requirements and/or the design, may arise in some of the distributed teams [20]. If these ambiguities are not solved in time it may cause long delays, leaving teams idle and frustrated, and reducing final quality of the project. Furthermore threats, threat of opportunism, security [21], trust, cultural issues and languages barrier are other factors which make the coordination management more difficult and challenging. The communication issues (e.g., distance, time zone difference, infrastructure support, distinct backgrounds, and lack of informal communication) may cause the loss of control over the teams located far away. If the company appoints a manager at the remote site and manager is not from their own vicinity, the manager might face a lot of problems. The employees behave as a “loose cannon” or shows excessive defensiveness or negativity [22]. The employees also feel these managers as a micromanager or ‘put on the spot’. In fact these reflections of employees towards the manager may occur due to differences in intercultural factors [23].

One major issue, which is responsible for controlling the quality of product, is the criteria for the distribution of the tasks amongst several companies or teams involved in the development of single project. So deciding which task is given to which company and what should be the criteria for selection of those companies distributed around the world is a difficult problem to be solved. However, the answer of the question might be very simple. Based on the expertise of the company or the developers, one manager can decide. But, is this sufficient? For example, if one outsourcing company is in India then should the testing be performed there or should be at customers site? [24] These are examples of questions that, if not properly solved, would directly affect the low quality of the product. Further, less obvious issues in task assignments also influence the quality of product. For example, an assignment decision may fail due to a high staff turnover rate. If people leave the company every year, then you get no knowledge and no return on investment.

Cross cultural issues [23], [25], [26] impact heavily in achieving quality objectives. MacGregor et al. [24] identified that intercultural factors [27] manifest themselves on a variety of levels in technical professions, all of which potentially impact the success of a project in GSD. The authors [24] gave examples of several researchers’ works and argued that the intercultural factors impact very seriously in GSD projects. Neglecting these factors may cause low quality of process/development and, therefore may cause the failure of the project. Several intercultural factors were identified, e.g., need for cross-cultural sensitivity [28], communication challenges in mediated communication [29], difficulty with planning and management of global innovation [30], differences in work-style [31], and power, hierarchy and agency [32] These all factors directly or indirectly impact on the quality of process and projects. In a case study of a GSD project distributed in three continents: North Europe, Asia and South America. Gibbs [33] observed that the global team was loosely coupled due to team members’ multiple cultural identifications, geographical dispersion, time differences and electronic rather than face-to face communication [33]. The loose coupling may be advantageous up to some extent but it seriously affects on the quality of process and product.

If we organized all the above factors, which are responsible for quality issues in development process and final product, we can conclude that the quality is affected by two major challenges: Requirement and Coordination. The factors affecting these two challenges can be summarized as follows:

Requirements challenges:

1. Communication: affected by several sub factors:
1.1 Distance between stockholders and team members.
1.2 Languages ([34]).
1.3 Culture and time zone differences [34].
2. Ambiguity in understanding the requirements.

Coordination Challenges:

Coordination challenges are affected by the following factors:

1. Communication [35][36].
2. Lack of Trust [37].
3. Intercultural issues [38].
4. Work allocation assignments [24].
5. Ambiguity in understanding:
    a. Organizational process,
    b. Management practices
    c. Requirement and design
6. Project planning and follow up [30]
7. Loss of control

 ISBN: 978-1-61208-165-6

IV. POSSIBLE SOLUTIONS

In the previous sections, we have shown the factors which are responsible for increasing the complexity in achieving the quality in both process and product in GSD. In this section, we are suggesting possible solutions, which are extracted from various sources. In fact, the different solutions available in literature for the problems of GSD are not normally concentrated on the quality aspect. In this paper, we have concentrated only on those ones which impact on quality aspects of GSD. We have indentified them and then unified and presented as possible solutions for quality issues in GSD. Further, it is observed that the quality of a software product is impacted by most of the problems of GSD. However, we do not find any paper in which all those issues are unified for the evaluation of both the quality of process and product. The following list is the results of our work:

1. QA strategies should be applied: To control the quality in software process and product in GSD environment, these QA strategies should include a quality management network, change control, active risk management, quality audits, inspection strategy, delivery strategy and reporting and measurement [4]. The authors have applied these practices in GSD environment and tested on a real project [4] and found that these are very effective in achieving quality in both process and project.

2. Well defined process. The type of development processes should be clearly defined. Requirements should be presented in such a way that it can be understood easily. One of the ways for clear understanding of requirement is modeling. Furthermore, the architecture design and its dependencies should also be elaborated with full care. Components and their interfaces should be well and precisely defined. Assignment of the task should be clear-cut to independent teams. There should not be any ambiguity in the task assignment. At remote sites, some particular person should be accountable for any query and conversations from outside.

The coordination and requirement challenges can be overcome adopting the following practices:

1. Proper planning and scheduling of all activities as a way to overcome the coordination and requirement problems [39][40].

2. Near shoring. One of the major problems in cross cultural environment is the behavior of people which varies from place to place and also increases the perception of lack of trust. Development should be done as near to the customers as possible i.e., it should be in close proximity of the parent company. It will reduce the cultural and language related problems. In this way, we can also solve the problems related to requirement challenges. Due to common/similar languages and close time zone, the development team can interact with customer(s) not only more frequently but also understanding the customers' requirements and problems in more fruitful way. Some of the typical examples for near shoring are: South Korea, Eastern European counties, Middle and South American countries as recommendable places for outsourcing for China, West European countries, and US respectively [41][42]. The sites in the same time zone are better options [34]. Of course, this may also be solved by extending the working hours at the different sites [43].

3. Strategic choice. Only those components which are not cultural sensitive should be outsourced reducing the risk of cross-cultural problems [25]; e.g., middleware or a component to be embedded in an Operating System (OS). In fact, the software for the middleware and for OS are not strongly affected by the surrounding environment and possible cultural issues because generally independent software and the specifications are clear from the beginning. Another suggestion was to outsource only those projects whose benefits are expected to outweigh the risks [23].

4. Transfer of knowledge and people. In [25], it is suggested that acceptance of the project in distributed environments should be made on the basis of some transfer of knowledge in exchange for the cultural risks. The quality suffers, if the staff at remote sites is not knowledgeable or expert. The authors also suggested that effective in-depth working relationships should be achieved amongst the development team throughout the project. Another suggestion for reducing the negative impact of intercultural factors is to create a "negotiated culture" [3]. Furthermore, a small number of the staff members should also be trained in other sites language [34]. Some staffs should also be exchanged for short periods of time. This practice will dramatically improve the working environment [11].

5. Synchronization in organization, processes, or technology in all the locations of the project is an effective technique for improving collaboration [44].

V. DISCUSSION

The acceptance of outsourcing in software development is increasing very rapidly. Only in India, Outsourcing to India has been increased 10 times in 10 years (1998-2007) and it is expected to increase on the annual rate of 28 percent for IT projects [45]. Additionally, not only outsourcing but global in-sourcing (i.e., development with own subsidiaries in foreign countries [24]) is also growing rapidly. For example, IBM has increased its staff in India from 53,000 to 73,000 in just one year in 2007 [46]. Naturally, if the rate of outsourcing is increasing to fulfill the demand of market, several new companies are also entering to the software market. However, it is not easy to control the quality at remote sites. Besides several other reasons, evaluating the capability and the talents at remote sites [47] are challenging job [47]. Additionally, it is not unusual to observe the lack of experts at the remote site: even they exist, it is usual that the number is not sufficient to fulfill what it is required. It is observed through surveys and studies, it is observed that these are the reasons why; the work assigned at distributed sites normally takes significantly longer time than what it is usual in other environments [35]. Again, the challenges for the requirements management and the project coordination are the causes. Further, several factors which contribute to these challenges (e.g., communication [35] [36], lack of trust [27], intercultural factors, lack of qualified and experienced professionals etc.) reduce the quality of the process and

 ISBN: 978-1-61208-165-6

product. These all factors increase the costs and reduce the probability of success.

Many works have been carried out by different researchers (e.g., [48], [49], [50]) to identify the problems, issues or challenges and solutions for GSD. If we analyze the majority of the available literature, we can easily find out that, only few of them have given emphasis on the quality aspect of GSD. By keeping all these issues, we conclude that there is still a need of investing more thinking and work to establishing procedures and methods for achieving the quality product in distributed environments.

VI. CONCLUSIONS

In this paper, we have focused more particularly into quality aspect of the product in global distributed environments. The quality of a software product developed in GSD environment is heavily impacted by problems of this type of environments. We have identified two major challenges responsible of causing most of the problems: requirements management and project coordination. Several issues and several factors which are responsible for these two challenges have been also suggested. We have shown how these challenges finally affect the quality of the product. Since all mentioned issues can become serious obstacles to quality of software products and they are not properly researched yet in point of view for quality aspects in GSD, our work may create attract the attention of the academic community towards this important issue. In this sense we consider this paper as a valuable contribution to the software engineering community.

REFERENCES

[1] D.W. Karolak, Global Software Development: Managing Virtual Teams and Environments, New York: Wiley-IEEE Computer Society Press, 1998.

[2] C. Bartelt, M, Broy, C., Herrmann, E., Knauss, M., Kuhrmann, A., Rausch, B., Rumpe, and K. Schneider, "Orchestration of Global Software Engineering Projects - Position Paper" Proceedings of the 2009 Fourth IEEE International Conference on Global Software Engineering (ICGSE '09). IEEE Computer Society, Washington, DC, USA, 2009, pp. 332-337.

[3] G.M. Olson and J.S. Olson "Distance Matters" Human-Computer Interaction, 15, 2000, pp. 139-178.

[4] M. Ivček and T. Galinac, "Aspects of Quality Assurance in Global Software Development Organization", Proceedings of the 27th International Conference on Telecommunications and Information of the 31th International Convention MIPRO 2008, 2008, pp. 150-155.

[5] O., Gotel, V., Kulkarni, M., Say, C. Scharff, and T. Sunetnanta, "Quality Indicators On Global Software Development Projects: Does 'Getting To Know You' Really Matter?" Proceedings of the 2009 Fourth IEEE International Conference on Global Software Engineering (ICGSE '09), 2009, pp. 3-7.

[6] IEEE, IEEE Std 610.12-1990-IEEE Standards Glossary of Software Engineering Terminology , IEEE Computer Society, 1991.

[7] D. Galin, Software Quality Assurance, From theory to implementation, Pearson Addison Wesley, 2004.

[8] P.B. Crossby, Quality is free, New York: McGraw-Hill, 1979.

[9] J.M. Juran, Juran's Handbook Quality control, New York, McGraw Hill, 1988.

[10] R.S. Pressman, Software Engineering, NewYork: McGraw-Hill, 2008.

[11] J.D. Herbsleb, "Global Software Engineering: The Future of Socio-technical Coordination". Proceedings of the 2007 Future of Software Engineering (FOSE '07), 2007, pp. 188-198.

[12] M., Lormans, H., van Dijk, A., van Deursen, E. Nöcker, and A. de Zeeuw, "Managing Evolving Requirements In An Outsourcing Context: An Industrial Experience Report". Proceddings of the Int. Workshop on Principles of Software Evolution (IWPSE'04), 2004 , pp. 149-158.

[13] M. M. Lehman, "Software's future: Managing evolution" IEEE Software, vol. 15, 1998, pp. 40-44.

[14] J., García Guzmán, J., Saldaña Ramos, A., Amescua Seco and A. Sanz Esteban, "Success Factors for the Management of Global Virtual Teams for Software Development", International Journal of Human Capital and Information Technology Professionals, vol. 2, 2011, pp. 48-59.

[15] A. Cockburn, and J., Highsmith, "Agile Software Development:The People Factor", Computer, vol. 34, November 2001, pp. 1-3

[16] ISO, ISO/IEC 9126-1Software Engineering - Product quality - Part 1: Quality model, Geneve: International Standards Organization, 1997.

[17] IEEE, IEEE std. 730-2002 Standard for Software Quality Assurance Plans, Washington: IEEE Computer Society, 2002.

[18] A. Abram and J. W. Moore, Guide to Software Engineering, Body of Knowledge, Washington: IEEE Computer Society, 2004,.

[19] R. Ahmad, Analysing Requirements Issues in Global Software Development, Dissertation, Honours Programme of the School of Computer Science and Software Engineering, The University of Western Australia, 2010.

[20] R., Sangwan, M., Bass, N., Mullick, D.J.Paulish, and J. Kazmeier, Global Software Development Handbook, Boston: Auerbach Publications, 2007.

[21] P. Mohagheghi, Global Software Development, Inssue , Solution And Challenges, Trial lecture, Dept. Computer and Information Science (IDI), University of Science and Technology (NTNU),Trondheim, Norway, 2004.

[22] L. Laroche, Managing Cultural Diversity In Technical Professions. Burlington: Butterworth-Heinemann, 2002,.

[23] G. Hofstede, Culture's consequences. 2nd Ed., Thousand Oaks: Sage, 2001.

[24] A. Lamersdorf, Towards a Global Software Development Distribution Model: Empirically-based Model Building for Distributed Software Development, Master thesis, University of Kaiserslautern, 2008.

[25] S, Krishna, S. Sahay, and G. Walsham, "Managing cross-cultural issues in global software outsourcing", Communications of the ACM, vol. 47, 2004, pp. 62-66.

[26] G. Borchers, "The software engineering impacts of cultural factors onmulti-cultural software development teams", Proc. of 25th Intl.Conference on SW Engineering,, 2003, pp. 540-545.

[27] P. Kruchten, "Analyzing intercultural factors affecting global software development – A position paper", Proceedings of GSD2004 3rd International Workshop on Global Software Development, 2004, pp. 59-62.

[28] M. L. Maznevski and K. M. Chudoba, "Bridging Space Over Time:Global Virtual Team Dynamics and Effectiveness", Organization Science, vol. 11, 2000, pp. 473-492.

[29] S. Whittaker, "Theories and Methods In Mediated Communication" In The Handbook Of Discourse Processes, Graesser A.C., Gernsbacher M.A. and Goldman S.R. (eds), LEA, Mahwah, NJ. 2003, pp. 243-286.

[30] P. Banerjee, "Narration, Discourse and Dialogue: Issues in theManagement of Intercultural Innovation", AI & Society, vol. 17, 2003, pp. 207-224.

[31] G. Walsham, Globalization and ICTs: Working across cultures, WP 8/2001 Research Papers In Management Studies, University of Cambridge, 2001.

[32] B.Nicholson, and S. Sahay, Some Political And Cultural Issues In The Globalisation Of Software Development: Case Experience From Britain and India, Information and Organization, vol. 11, 2001, pp. 25-43.

[33] J. L. Gibbs, Loose coupling in global teams: tracing the contours of cultural complexity, Ph. D. dissertation, University of Southern California, Los Angeles, 2002.

[34] E. Carmel, and R. Agarwal, "Tactical Approaches for Alleviating Distance in Global Software Development.", IEEE Software, vol. 18, 2001, pp. 22-29.

[35] J.D., Herbsleb, A., Mockus, T.A. Finholt, and R.E. Grinter, "An Empirical Study Of Global Software Development: Distance and Speed" Proceedings, International Conference on Software Engineering, Toronto, Canada, 2001, pp. 81-90.

[36] J.D. Herbsleb and A. Mockus, "An Empirical Study of Speed and Communication in Globally-Distributed Software Development". IEEE Transactions on Software Engineering, vol. 29, 2003, pp. 481-494.

[37] N. B. Moe and D. Smite, "Understanding a Lack of Trust in Global Software Teams: A Multiple-Case Study", Software Process, vol. 13, 2008, pp. 217-231.

[38] G. Hofstede, Culture and organization – software of the mind. 3rd Ed., New York: McGraw-Hill, 2010.

[39] H. Kerzner, Project Management: A System approach to Planning, Scheduling and Controlling, New York: John Wiley & Sons, 2003.

[40] PMI, PMBOOKA Guide to the Project Management Body of Knowledge (PMBOK Guide), Pennsilvania: Project Management Institute, 2004.

[41] D. Dubie, "Outsourcing Moves Closer to Home", CIO Today, December 18th, 2007.

[42] C. Zarley, "7 Reasons the Bloom is Off Asian Outsourcing". The Channel Wire, December 12th, 2007.

[43] G. Lee, W. DeLone, and J. A. Espinosa, "Ambidextrous Coping Strategies In Globally Distributed Software Development Projects", Communications of the ACM, vol. 49, 2006, pp. 35-40.

[44] R., Heeks, S., Krishna, B. Nicholson, and S. Sahay, "Synching or Sinking: Global Software Outsourcing Relationships". IEEE Software, vol. 18, 2001, pp. 54-60.

[45] M. Kobayashi-Hillary, "India faces battle for outsourcing" BBC news, August 17th, 2007, http://news.bbc.co.uk/2/hi/business/6944583.stm, Last accessed August 2011.

[46] J. Ribeiro, "IBM expects $1 billion in India revenue this year", InfoWorld, December 17th , 2007.

[47] W. Kobitzsch, H.D. Rombach, and R.L. Feldmann, "Outsourcing in India", IEEE Software, vol. 18, 2001, pp. 78-86.

[48] J.D. Herbsleb and D. Moitra, "Global software development", IEEE Software, vol.18, 2001, pp.16-20.

[49] J. D. Herbsleb, D. J. Paulish, and M. Bass, "Global software development at siemens: experience from nine projects", Proceedings of the 27th international conference on Software engineering (ICSE '05). 2005, pp. 524-533.

[50] R. Prikladnicki, J.L. Nicolas Audy and, R. Evaristo, ``Global software development in practice: lessons learned''. Software Process: Improvement and Practice, vol. 8, 2003, pp. 267-281.

 ISBN: 978-1-61208-165-6