ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
UDC/UDK 004.413.896:65.012.4

A DISCUSSION ON THE ROLE OF PEOPLE IN GLOBAL SOFTWARE DEVELOPMENT

Sanjay Misra, Ricardo Colomo-Palacios, Tolga Pusatli, Pedro Soto-Acosta

Preliminary notes
Literature is producing a considerable amount of papers which focus on the risks, challenges and solutions of global software development (GSD). However, the influence of human factors on the success of GSD projects requires further study. The aim of our paper is twofold. First, to identify the challenges related to the human factors in GSD and, second, to propose the solution(s), which could help in solving or reducing the overall impact of these challenges. The main conclusions of this research can be valuable to organizations that are willing to achieve the quality objectives regarding GSD projects.

Keywords: *global software development, people management, software industry, People CMM*

Rasprava o ulozi ljudi u globalnom razvoju softvera

Prethodno priopćenje
U literaturi se može naći priličan broj radova koji se bave rizicima, izazovima i rješenjima za globalni razvoj softvera (GSD). Međutim, utjecaj ljudskog faktora na uspjeh projekata o globalnom razvoju softvera zahtijeva dodatno istraživanje. Ovaj rad ima dvojak cilj. Prvo, identificirati izazove povezane s ljudskim faktorima u GSD i, drugo, predložiti rješenje (rješenja) koje bi moglo pomoći u rješavanju ili reduciranju cjelokupnog djelovanja tih izazova. Glavni zaključci ovog istraživanja mogli bi biti važni organizacijama koje žele postići kvalitetne rezultate projekata koji se bave globalnim razvojem softvera.

Ključne riječi: *globalni razvoj softvera, upravljanje ljudima, softverska industrija, People CMM (People Capability Maturity Model - Model osposobljavanja ljudi)*

1 Introduction

Software development from A to Z is rarely the result of one person's work only. Today, software development is a team work. Furthermore, with the booming of the Internet and the adoption of electronic communication systems, the semantic of this team work has become more and more important. As defined by Sangwan et al. [1], global software development (GSD) is the development of software through teams, from multiple geographic locations, that may pertain to the same organization or to other collaborating companies.

GSD teams have evolved from a single site to a multiple localization working environment [2]. As a result, firms developing and/or maintaining software products cannot ignore the impact of GSD [3], since it is driving a deep transformation in the way that products are conceived, designed, constructed, tested, and delivered to customers [4]. The final result of this process is that software development is becoming a multi-site, multicultural as well as globally distributed undertaking (e.g. [5, 6]). Although GSD is a de facto tendency in today's IT industry, sometimes it is criticized for being slow and hindering. In spite of this, nowadays software products are developed collaboratively in multiple locations around the world. Projects are being contracted in whole or in part [7] with several motivations, including the desire of being close to local markets. However, this motivation is not enough to enable GSD to work as fast as traditional team work, where everyone is in the same building. One of the most recognized benefits with regard to the adoption of e-communications is that, as long as employees are connected, they can work no matter where they are. However, an empirical study by Herbsleb and Mocus [8] reported that a distributed setting can take 2.5 times longer to do similar tasks when compared to a non-distributed setting. It is so because GSD may be contradictory to agile development, a widely accepted practice in software developments. One of the reasons for the success of agile developments is that people are placed closer together, so that teams can be more effective. This practice reduces the cost of moving information, employees can talk, discuss and solve problems immediately [9]. In contrast, GSD places people around the globe and, therefore, the agility goal is hardly achieved. Although agile development cannot be set up along the entire project the nodes of software development centres could adopt it.

The importance of GSD management has led to a huge effort in the art and science of organizing and managing globally distributed software development. However, there is still a need for further research regarding the development of methods, techniques and practices before GSD can be considered a mature discipline [10], since the globalization of software development introduces a great deal of complexity in an already complex process [11].

Human resources management is a key issue in any software development project, including GSD projects [12]. The importance of human resources in software engineering was confirmed more than a decade ago, when Software Engineering Institute (SEI) developed a separate model for personnel management: the people management capability maturity model (PM-CMM) [13]. More recently, several studies have been devoted to shed some light into people management aspects and GSD environments [14 ÷ 21]. In this complex scenario, the aim of this paper is double. Firstly, it is aimed to identify the challenges related to the human factors in GSD and, secondly, to propose the solution(s), which could help in solving or reducing the overall impact of these challenges.

The remainder of the paper is organized as follows.

The next section introduces personnel issues in software development projects. In section 3, challenges are identified. Following that, observations and solutions related to personnel management in GSD projects are discussed in section 4. Then, section 5 describes recommendations for human resources management within GSD projects and, finally, conclusions are drawn in section 6.

2 The personnel issues in software projects

Software project management is a relatively recent discipline that emerged during the second half of the 20th century. The task of managing a software project can be extremely complex and the reasons for such complexity may arise from personal, team and organizational resources [22]. Software project management involves scheduling, planning, monitoring, and controlling personnel, processes, and resources to achieve specific objectives, whilst at the same time satisfying a variety of constraints [23], such as limited resources. Erdogmus [24] suggested that software processes can be placed inside a triangular map according to their emphasis with regard to three aspects: people, technology and rigor. The human dimension is key in software engineering and software development is an intense human capital activity, more intense in intellectual capital [25]. Consequently, the human and social aspect of software engineering has turned into an important topic to investigate for both scholars and practitioners who strive to improve organizational efficiency [26]. Although the importance of human resources has been widely recognized as key for software engineering, there is wide acknowledgment that suggests researchers should put a larger focus on humans factors involved in software engineering [27].

In global software development environments, literature reported some interactions with agile methodologies and successful implementations. In such methodologies, the individual competence is the main success factor. In other words, agile methods put more emphasis on the people factors [9]. Thus, agile development focuses on individuals' talents and skills, adapting the process to specific people and teams. Several important and recent studies are devoted to analyse the interactions among agile teams and people factors [28], [29] along with its evolution over time [30].

However, agile methods are not alone in people factors research. Other important topics in software development research, related to the importance of people in software development/maintenance are: assigning people to specific roles [31], [32], productivity issues in software engineering and IT projects [26], [33], [34], skills identification [25], relationship issues [35] and emotions[36] citing the most important and recent studies.

3 Personnel related challenges in GSD projects

As mentioned in the introduction, GSD may cause a profound impact on the product generation from the planning phase to its deployment. In this sense, ignoring the management of such teams might bear risks. Additionally, software development in a GSD context may increase this complexity significantly with respect to communication, coordination and control issues [37]. In this work, authors suggested that a major challenge within GSD teams is the lack of informal communication. In this same line, Herbsleb [38] asseverated that the key phenomenon in GSD teams is the coordination over distance.

To address the critical success factors in GSD, quasi descriptive/explanatory models have been developed. For instance, Sangwan et al. [1] introduced a framework to coordinate requirement engineering, architecture design, project planning and product development, itemizing critical success factors such as ambiguity reduction, stability maximization, dependencies understanding, coordination and balance between flexibility and rigidity. However, even though these factors are important for all software projects whether GSD is adopted or not, the mentioned factors gain more importance with distance among teams.

Other researchers (e.g. [17]) suggest that the added complexity of GSD over traditional software development comes from:

a) Lack of common understanding of goals and requirements assigned.
b) Difficulties in communication (members are geographically separated).
c) Bottlenecks and problems in project execution (Variety of processes, management mechanisms, and associated skills/competencies).
d) Ineffective management of knowledge sharing.

All these challenges are related mainly to people and their influence on software development. Prikladnicki et al. [6] through a case study identified seven aspects related to GSD projects, where difficulties may arise: requirements engineering; software development process; software configuration; knowledge management; communication and language; culture; context sharing and trust. According to the authors, requirements engineering is the main challenge for the software development process point of view.

More recently, Jiménez et al. [39] conducted a systematic literature review and synthesized ten main challenges regarding GSD: communication; group awareness; software configuration management; knowledge management; coordination; collaboration; project and process management; process support; quality and measurement; and risk management.

Communication bears great importance in software projects in general and in GSD in particular. However, when it comes to discuss communication, it should not be confined and/or generalized as the exchange of data/information or knowledge. Communication is beyond the transmission of messages. Saray et al. [40] reported the communication challenges encountered in a case study that led to the structuring of the business model, the project management practices and the development of social relationships.

Cultural complexities play a crucial role in GSD teams. Not in vain, disperse teams pertaining to different countries and with different backgrounds are meant to work together. The operation of globally distributed software development projects requires a level of

cooperation and coordination that cannot ignore the impact cultural diversity plays and the barriers and misunderstandings it can and does create [41]. GSD poses a challenge to work practices in the form of miscommunication and misinterpretation of shared tasks. These conflicts and misunderstanding arise unless people learn how to interact in a harmonic way with persons from different cultures [16]. Given that, organizations must develop the workers' intercultural skills in order to adapt their workforce to this new scenario.

Finally, temporal distance is another barrier to collaboration and communication. GSD teams are placed in different time zones and the main problem with having developers working in different time zones is that there are fewer hours in the work day when multiple sites can participate in synchronous meeting [42].

Tab. 1 summarizes the main challenges of GSD with respect to its human dimension in the literature.

Table 1 GSD Challenges related to personnel

Challenge	Literature support
Communication	[6], [37], [39], [43], [44]
Knowledge Management	[6], [37], [39], [45]
Coordination	[37 ÷ 39], [43], [46]
Collaboration	[3], [39], [43], [47 ÷ 49]
Socio-Cultural distance (Lack of group awareness)	[6], [50 ÷ 54]
Lack of trust	[2], [55 ÷ 57]

4 Observations, discussions and suggestions

Taking into account the literature review, three main observations can be drawn:

1. Challenges related to people are important in GSD as stated above. However, in spite of its importance, neither practitioners nor researchers have given enough attention to this phenomenon. Majority of the organization in software industry give more emphasis on tools and technology and little on people [58]. Fernandez and Misra [59] stated that it is strange that human and social factors, which are related to people, affecting development teams have attracted little attention.
2. Most of the companies involved in GSD hire personnel at low costs for fast output/delivery of projects/sub projects. Fast delivery at low cost, one of the main advantages of GSD, is however responsible for most of the challenges related to people in this new working environment. Staff turnover in Asian countries and attrition in west Europe is comparatively higher [60]. These authors listed insufficient competence, wage and staff turnover are among top ten risks of global software engineering.
3. People are the most important component in software development. However, especially in small and medium-sized software companies that employ GSD, an employee-care culture is almost nonexistent and, thus, temporary contracts, low personal development perspectives, low salaries and so on are very frequent… In such scenario, talented personnel will not stay long and the personnel continuity has been suggested as a factor that increases quality and effectiveness of software development. In fact staff turnover is a generic risk and occurs when global software engineering (GSE) has no clear integration with an organization's strategy and carrier paths [60]. Ebert et al. [60] identified talent as one of drivers of success for GSE. However, talents may move other places due to several factors such as level of job satisfaction, retaining and encouragement policy of employees, working conditions & workload [61]. The result will be the high risk of failure of GSE [60].

Tab. 1 provides a good starting point to setup a research strategy to itemize factors to facilitate people related challenges. These factors are the preliminary challenges that should be tested empirically. In addition to them, other factors such as cultural and language misunderstandings, political barriers and currency differences should be considered.

Taking into account the challenges identified and previous observations observed by the authors who were/are involved in different GSD projects in EU, Turkey and Argentina, the following solutions to these issues can be drawn:

1. To Improve Communication. The problem of communication arises due to differences in language. This is a very common problem among the software developers especially in those countries where English is not an official and common language. Possible solutions to this problem are:
   a. Written communication means should be the preferred way of communication in comparison with verbal communication. Authors personal experience is that most of software developers (especially in non-English speaking countries e.g. Turkey, Spain) can better understand the problem if it is written. This gives the opportunity to think or read twice, a chance that verbal communication does not provide. Given that differences in language proficiency among distributed team members create barriers to effective communication [42, 62] and that because of linguistic barriers software engineers prefer asynchronous mode of communication such as e-mails [1], a more formal communication would lead to a better common understanding. For instance, E-mail was the recommended means of communication between central and remote teams [63].
   b. There should be experts at each node of development centres, who can understand both languages, i.e. English and the local language. Certainly, these people can be used to help with communication problems.
   a. When selecting GSD members, a good command of English by software developers should be preferable. It is important to note here that adopting this practice does not necessarily mean that if a non-English speaker expert person needs to be included in the team, he/she will be dismissed just because of the lack of English speaking skills.
2. Knowledge management, competence management and performance appraisal. According to Ball and Harris [64] IT personnel evaluation is the second

most critical issue in IT management. GSD scenarios make this problem even worse. A possible solution to this problem could be to set up contracts with companies certified in both CMMI and People-CMM. In [12], authors investigated the applicability of different process areas from the People-CMM within the GSD context; this investigation leads to the conclusion that competence management and performance appraisals are feasible objectives when dealing with People-CMM and GSD joint.

3. To improve coordination and collaboration. The communication problem stated previously also has a negative impact on coordination and collaboration [65]. Moreover, cultural differences may negatively influence coordination and collaboration [66]. Taking this into account, solutions suggested for communication problems may be also applicable to reduce this problem. Apart from those, other possible solutions are:
   a. Work division. That is, distributing the work among individuals instead of a group. Before task allocation, it should be fully confirmed that the proper knowledge is available on the site and the person is capable to perform the task. Although this solution is more effective at small-scale GSD projects, it could be also adopted in larger projects. Atomizing the work provides good traceability for the person-software artefact couple, a key issue in software engineering. Coordination is lower due to the distribution of loosely coupled work packages to different sites and workers but, in the other hand task architecture can be fixated in advance and that uncertainty is limited [67].
   b. Verbal communication should be standardized and if it becomes a necessity, video conferencing instead of phone calls should be used [68]. Nonetheless, these discussions should be documented and sent to the proper person(s) for avoiding any ambiguity, that, in any case arises when dealing with multicultural teams [41].
4. To reduce Socio-Cultural distance (Lack of group awareness). Socio-cultural differences are a very common problem in GSD. It may occur in different sites of the same countries or different sites in other countries. However, if all the workers are using English, the effect of socio-cultural difference is less important, however, team members with more proficient language skills may lack confidence in their remote counterparts' understanding of communication [42]. Some tools devoted to build group awareness are presented in [39]. Authors believe that, given that the lack of group awareness leads to coordination breakdowns [69] and lack of trust, these two issues must be faced in a joint effort.
5. To increase trust. Trust is especially vital in GSD teams due to the lack of face-to-face interactions. The lack of trust between team members may affect their contribution and reduce the transfer of information between members. Also, it may move individuals to pursue personal goals rather than group goals, make them feel the need to double check work performed by others and the quality and productivity of their work could decrease to lower levels. To avoid the negative consequences of this factor, authors suggest the following actions:
   a. To design and implant a formal trust building process (i.e. [2]).
   b. To organize workshops, joint trainings or just invest in several face-to-face meetings especially at the beginning of the project. This event can give the opportunity to discuss relevant aspects of the project, teach necessary technologies and develop personal relationships [42, 70, 71].
   c. To promote continuity in partnerships. Long term partnerships encourage trust and common knowledge and the building of the "third culture". This will lead to higher trust and a better work package allocation [5].
   d. To promote continuity in software personnel. Collaborative software development projects need more resource continuity than outsourcing projects [72]. One of the traditional assumptions in software industry is that software personnel continuity leads to smaller costs [73]. Given that, authors suggest to adopt policies to ensure higher continuity for software practitioners also in GSD projects.

5 Recommendations

Based on the above discussion, experience and literature review, three main recommendations are drawn in order to improve the quality of the product produced through a GSD environment.

1. Invest in people
   a. Select talented staff
   b. Provide good job conditions
   c. Pay employees according to market salaries
   d. Promote a rewarding strategy for personnel (recognition, reward…)
   e. Promote a transparent financial status
   f. Finance employee development programs including mentoring, coaching…
2. Promote the mobility of software development team members
   a. Design a long term mobility program
   b. Design a project-scope mobility program
   c. Promote the study of foreign languages and cultures.
3. Improve GSD processes
   e. Adopt software and process improvement initiatives and models (CMMI).
   f. Adopt people improvement initiatives and models (People-CMM).

6 Conclusion

One of the major drawbacks of global software development is that low quality software can be produced. Many factors lead to this problem, i.e. communication, knowledge management, coordination, collaboration, group awareness trust… Many of these factors are related to people. This is not surprising since people perform

software projects and are responsible for the overall quality of software artefacts. Considering this, it is undeniable that GSD must be aware of people issues. As a consequence result, investing in people has become a major issue for GSD players, contractors and service suppliers.

This paper proposes some suggestions and recommendations in order to reduce the negative impact of people issues on GSD. However, future work will try to cover several other existing research gaps in the literature. First, the impact of competence management programs on GSD environments and, more precisely, on the overall quality of software products will be measured. Second, specific personnel performance metrics for GSD projects will be designed. Third, the influence of people and software process improvement practices on several aspects related to GSD and, more specifically, on trust and team awareness will be assessed. Finally, the impact of personnel mobility on GSD projects will be evaluated.

7 References

[1] Sangwan, R.; Bass, M.; Mullick, N.; Paulish, D. J.; Kazmeier, J. Global Software Development Handbook. Auerbach Publications, 2006.

[2] Hernández-López, A.; Colomo-Palacios, R.; García-Crespo, Á.; Soto-Acosta, P. Trust building process for global software development teams: a review from the literature. // International Journal of Knowledge Society Research, 1, 1(2010), pp. 65–82.

[3] Cusick, J.; Prasad, A. A Practical Management and Engineering Approach to Offshore Collaboration. // IEEE Software, 23, 5(2006), pp. 20–29.

[4] Herbsleb, J. D.; Moitra, D. Global software development. // IEEE Software, 18, 2(2001), pp. 16–20.

[5] Garcia-Crespo, A.; Colomo-Palacios, R.; Soto-Acosta, P.; Ruano-Mayoral, M. A qualitative study of hard decision making in managing global software development teams. // Information Systems Management, 27, 3(2010), pp. 247–252.

[6] Prikladnicki, R.; Nicolas Audy, J. L.; Evaristo, R. Global software development in practice lessons learned. // Software Process: Improvement and Practice, 8, 4(2003), pp. 267–281.

[7] Madachy, R. J. Cost modeling of distributed team processes for global development and Software-Intensive Systems of Systems. // Software Process: Improvement and Practice, 13, 1(2008), pp. 51–61.

[8] Herbsleb, J. D.; Mockus, A. An empirical study of speed and communication in globally distributed software development. // IEEE Transactions on Software Engineering, 29, 6(2003), pp. 481–494.

[9] Cockburn, A.; Highsmith, J. Agile software development, the people factor. // Computer, 34, 11(2001), pp. 131–133.

[10] Damian, D.; Moitra, D. Guest Editors' Introduction: Global Software Development: How Far Have We Come? // Software, IEEE, 23, 5(2006), pp. 17–19.

[11] Treinen, J. J.; Miller-Frost, S. L. Following the sun: Case studies in global software development. // IBM Systems Journal, 45, 4(2006), pp. 773–783.

[12] Colomo-Palacios, R.; Casado-Lumbreras, C.; Soto-Acosta, P.; Misra, S.; García-Peñalvo, F. J. Analyzing human resource management practices within the GSD context. // Journal of Global Information Technology Management, 15, 3(2012), pp. 30–54.

[13] Curtis, B.; Hefley, B.; Miller, S. People Capability Maturity Model (P-CMM) Version 2.0, Software Engineering Institute, 2009.

[14] Fernández, L.; Misra, S. Analysis of cultural and gender influences on teamwork performance for software requirements analysis in multinational environments. // IET Software, 6, 3(2012), pp. 167–175.

[15] El-Baz, H.; Zualkernan, I. A. Employee competency maturity model and its application in global software outsourcing. // International Journal of Computer Applications in Technology, 40, 3(2011), pp. 170–180.

[16] Casado-Lumbreras, C.; Colomo-Palacios, R.; Soto-Acosta, P.; Misra, S. Culture dimensions in software development industry: The effects of mentoring. // Scientific Research and Essays, 6, 11(2011), pp. 2403–2412.

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

[18] Hernández-López, A.; Colomo-Palacios, R.; García-Crespo, Á.; Soto-Acosta, P. Team Software Process in GSD Teams: A study of new work practices and models. // International Journal of Human Capital and Information Technology Professionals, 1, 3(2010), pp. 32–53.

[19] Tuffley, D. Optimising virtual team leadership in Global Software Development. // IET Software, 6, 3(2012), pp. 176-184.

[20] Ponisio, L.; van Eck, P. Metrics-based control in outsourced software development projects. // IET Software, 6, 5(2012), pp. 438-450.

[21] Gonçalves, M. K.; de Souza, C. R. B.; González, V. M. Collaboration, Information Seeking and Communication: An Observational Study of Software Developers' Work Practices. // Journal of Universal Computer Science, 17, 14(2011), pp. 1913–1930.

[22] Rose, J.; Pedersen, K.; Hosbond, J. H.; Kræmmergaard, P. Management competences, not tools and techniques: A grounded examination of software project management at WM-data. // Information and Software Technology, 49, 6(2007), pp. 605–624.

[23] Chang, C.; Christensen, M.; Zhang, T. Genetic Algorithms for Project Management. // Annals of Software Engineering, 11, 1(2001), pp. 107–139.

[24] Erdogmus, H. Essentials of Software Process. // Software, IEEE, 25, 4(2008), pp. 4 –7.

[25] Colomo-Palacios, R.; Tovar-Caro, E.; García-Crespo, Á.; Gómez-Berbís, J. M. Identifying technical competences of IT Professionals: the case of software engineers. // International Journal of Human Capital and Information Technology Professionals, 1, 1(2010), pp. 31–43.

[26] Yilmaz, M.; O'Connor, R. Social Capital as a Determinant Factor of Software Development Productivity. // International Journal of Human Capital and Information Technology Professionals, 3, 2(2012), pp. 40–62.

[27] Feldt, R.; Torkar, R.; Angelis, L.; Samuelsson, M. Towards individualized software engineering: empirical studies should collect psychometrics. // in Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering, New York, NY, USA, 2008, pp. 49–52.

[28] de O. Melo, C.; Cruzes, D. S.; Kon, F.; Conradi, R. Interpretative case studies on agile team productivity and management. // Information and Software Technology, 55, (2013), pp. 412–427.

[29] Hoda, R.; Noble, J.; Marshall, S. Self-Organizing Roles on Agile Software Development Teams. // IEEE Transactions on Software Engineering, PP, 99(2012), p. 1.

[30] Dingsøyr, T.; Nerur, S.; Balijepally, V.; Moe, N. B. A decade of agile methodologies: Towards explaining agile

software development. // Journal of Systems and Software, . 85, 6(2012), pp. 1213–1221.
[31] Ampuero, M. A.; Baldoquín de la Peña, M. G.; Castillo, S. T. A. Identification of Patterns for the Formation of Software Development Projects Teams. // International Journal of Human Capital and Information Technology Professionals, 1, 3(2010), pp. 69–80.
[32] Acuña, S. T.; Juristo, N. Assigning people to roles in software projects. // Software: Practice and Experience, 34, 7(2004), pp. 675–696.
[33] Yilmaz, M.; O'Connor, R. V. A software process engineering approach to improving software team productivity using socioeconomic mechanism design. // SIGSOFT Softw. Eng. Notes, 36, 5(2011), pp. 1–5.
[34] Rodríguez, D.; Sicilia, M. A.; García, E.; Harrison, R. Empirical findings on team size and productivity in software development. // Journal of Systems and Software, 85, 3(2012), pp. 562–570.
[35] Adam, A.; Bernard, W.; Thomas, M. Relationship Issues in Global Software Development Enterprises, 2008.
[36] Colomo-Palacios, R.; Casado-Lumbreras, C.; Soto-Acosta, P.; García-Crespo, A. Using the Affect Grid to Measure Emotions in Software Requirements Engineering. // Journal of Universal Computer Science, 17, 9(2011), pp. 1281–1298.
[37] Conchúir, E. O.; Holmström Olsson, H.; Ågerfalk, P. J.; Fitzgerald, B. Benefits of global software development: exploring the unexplored. // Software Process: Improvement and Practice, 14, 4(2009), pp. 201–212.
[38] Herbsleb, J. D. Global Software Engineering: The Future of Socio-technical Coordination. // in 2007 Future of Software Engineering, Washington, DC, USA, 2007, pp. 188–198.
[39] Jiménez, M.; Piattini, M.; Vizcaíno, A. Challenges and Improvements in Distributed Software Development: A Systematic Review. // Advances in Software Engineering, 2009 (2009), pp. 1–14.
[40] Sahay, S.; Nicholson, B.; Krishna, S. Global IT Outsourcing: Software Development across Borders. Cambridge University Press, 2003.
[41] Casey, V. Imparting the importance of culture to global software development. // ACM Inroads, 1, 3(2011), pp. 51–57.
[42] Noll, J.; Beecham, S.; Richardson, I. Global software development and collaboration: barriers and solutions. // ACM Inroads, 1, 3(2011), pp. 66–78.
[43] Kommeren, R.; Parviainen, P. Philips experiences in global distributed software development. // Empirical Software Engineering, 12, 6(2007), pp. 647–660.
[44] Fuller, M. A.; Hardin, A. M.; Davison, R. M. Efficacy in Technology-Mediated Distributed Teams. // Journal of Management Information Systems, 23, 3(2007), pp. 209–235.
[45] Mattarelli, E.; Gupta, A. Offshore-onsite subgroup dynamics in globally distributed teams. // Information Technology & People, 22, 3(2009), pp. 242–269.
[46] Ovaska, P.; Rossi, M.; Marttiin, P. Architecture as a coordination tool in multi-site software development. // Software Process: Improvement and Practice, 8, 4(2003), pp. 233–247.
[47] Siakas, K. V.; Balstrup, B. Software outsourcing quality achieved by global virtual collaboration. // Software Process: Improvement and Practice, 11, 3(2006), pp. 319–328.
[48] Striukova, L.; Rayna, T. The role of social capital in virtual teams and organisations: corporate value creation. // International Journal of Networking and Virtual Organisations, 5, 1(2008), pp. 103–119.
[49] Paasivaara, M.; Lassenius, C. Collaboration practices in global inter-organizational software development projects. // Software Process: Improvement and Practice, 8, 4(2003), pp. 183–199.
[50] Ali Babar, M.; Verner, J. M.; Nguyen, P. T. Establishing and maintaining trust in software outsourcing relationships: An empirical investigation. // Journal of Systems and Software, 80, 9(2007), pp. 1438–1449.
[51] Damian, D. Global software development: growing opportunities, ongoing challenges. // Software Process: Improvement and Practice, 8, 4(2003), pp. 179–182.
[52] Evaristo, J. R.; Scudder, R.; Desouza, K. C.; Sato, O. A dimensional analysis of geographically distributed project teams: a case study. // Journal of Engineering and Technology Management, 21, 3(2004), pp. 175–189.
[53] Krishna, S.; Sahay, S.; Walsham, G. Managing cross-cultural issues in global software outsourcing. // Commun. ACM, 47, 4(2004), pp. 62–66.
[54] Layman, L.; Williams, L.; Damian, D.; Bures, H. Essential communication practices for Extreme Programming in a global software development team. // Information and Software Technology, 48, 9(2006), pp. 781–794.
[55] Barczak, G.; McDonough, E. F.; Athanassiou, N. So You Want to Be a Global Project Leader? // Research-Technology Management, 49, 3(2006), pp. 28–35.
[56] DeRosa, D. M.; Hantula, D. A.; Kock, N.; D'Arcy, J. Trust and leadership in virtual teamwork: A media naturalness perspective. // Human Resource Management, 43, 2–3(2004), pp. 219–232.
[57] Oza, N. V.; Hall, T.; Rainer, A.; Grey, S. Trust in software outsourcing relationships: An empirical investigation of Indian software companies. // Information and Software Technology, 48, 5(2006), pp. 345–354.
[58] Mishra, A.; Misra, S. People management in software industry: the key to success. // SIGSOFT Softw. Eng. Notes, 35, 6(2010), pp. 1–4.
[59] Fernández-Sanz, L.; Misra, S. Influence of Human Factors in Software Quality and Productivity. // in Computational Science and Its Applications - ICCSA 2011, B. Murgante, O. Gervasi, A. Iglesias, D. Taniar, and B. O. Apduhan, Eds. Springer Berlin Heidelberg, 2011, pp. 257–269.
[60] Ebert, C.; Murthy, B. K.; Jha, N. N. Managing Risks in Global Software Engineering: Principles and Practices. // in IEEE International Conference on Global Software Engineering, ICGSE 2008, 2008, pp. 131 –140.
[61] Deepa, M.; Stella, M. Employee turnover in 'IT' industry with special reference to Chennai city-an exploratory study. // ZENITH International Journal of Multidisciplinary Research, 2, 7(2012), pp. 160–177.
[62] Lings, B.; Lundell, B.; Agerfalk, P. J.; Fitzgerald, B. A reference model for successful Distributed Development of Software Systems, 2007, pp. 130–139.
[63] Cataldo, M.; Bass, M.; Herbsleb, J. D.; Bass, L. On Coordination Mechanisms in Global Software Development. // in Second IEEE International Conference on Global Software Engineering, ICGSE 2007, 2007, pp. 71 –80.
[64] Ball, L.; Harris, R. SMIS members: a membership analysis. // MIS Q., 6, 1(1982), pp. 19–38.
[65] Misra, S.; Fernández-Sanz, L. Quality Issues in Global Software Development. // presented at the ICSEA 2011, The Sixth International Conference on Software Engineering Advances, 2011, pp. 325–330.
[66] Monasor, M. J.; Vizcaíno, A.; Piattini, M. Cultural and linguistic problems in GSD: a simulator to train engineers in these issues. // Journal of Software: Evolution and Process, 24, 6(2012), pp. 707–717.
[67] Mohan, S.; Fernandez, J. Distributed software development projects: work breakdown approaches to overcome key coordination challenges. // in Proceedings of the 3rd India software engineering conference, New York, NY, USA, 2010, pp. 173–182.

[68] Wiredu, G. O. Understanding the functions of teleconferences for coordinating global software development projects. // Information Systems Journal, 21, 2(2011), pp. 175–194.
[69] Bird, C.; Nagappan, N.; Devanbu, P.; Gall, H.; Murphy, B. Does distributed development affect software quality?: An empirical case study of Windows Vista. // Commun. ACM, 52, 8(2009), pp. 85–93.
[70] Lescher, C. Patterns for global development: how to build one global team? // in Proceedings of the 15th European Conference on Pattern Languages of Programs, New York, NY, USA, 2010, pp. 6:1–6:6.
[71] Moe, N. B.; Šmite, D. Understanding a lack of trust in Global Software Teams: a multiple-case study. // Software Process: Improvement and Practice, 13, 3(2008), pp. 217–231.
[72] Forbath, T.; Brooks, P.; Dass, A. Beyond Cost Reduction: Using Collaboration to Increase Innovation in Global Software Development Projects. // in IEEE International Conference on Global Software Engineering, ICGSE 2008, 2008, pp. 205–209.
[73] Colomo-Palacios, R.; Fernandes, E.; Soto-Acosta, P.; Sabbagh, M. Software product evolution for Intellectual Capital Management: The case of Meta4 PeopleNet. // International Journal of Information Management, 31, 4(2011), pp. 395– 399.

Author's addresses

Prof. (Dr.) Sanjay Misra
Atilim University, Department of Computer Engineering
Ankara, Turkey
smisra@futminna.edu.ng

Dr. Ricardo Colomo-Palacios
Universidad Carlos III de Madrid, Computer Science Department
Av. Universidad 30, 28911 Leganés, Madrid, Spain
ricardo.colomo@uc3m.es

Dr. Tolga Pusatli
Department of Mathematics and Computer Science
Cankaya University**,** Ankara, Turkey
pusatli@cankaya.edu

Dr. Pedro Soto-Acosta
Universidad de Murcia, Department of Management & Finance
Campus de Espinardo, 30100 Espinardo, Murcia, Spain
psoto@um.es