JOURNAL OF SOFTWARE: EVOLUTION AND PROCESS
J. Softw. Evol. and Proc. 2012; **24**:119–137
Published online 5 August 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/smr.475

Propinquity in global software engineering: examining perceived distance in globally distributed project teams

Rafael Prikladnicki*,†

Computer Science School, PUCRS, Porto Alegre, Rio Grande do Sul, Brazil

SUMMARY

In social psychology, propinquity refers to the physical (objective) or psychological (subjective) proximity between people. In this paper, we explore the psychological dimension of propinquity by examining the phenomenon of feeling distant from geographically distributed people, in the context of distributed software teams. The perceived distance is an important challenge faced by distributed teams, and it is frequently based on factors beyond the physical distance, such as communication and cultural differences. The purpose of this paper is to present a model to assess and make more visible the construct of 'perceived distance' among members of global software engineering teams. The model was applied in three real-world cases to assess its effectiveness in uncovering hidden and useful information during the project lifecycle. The practical experience lived with the quantification of perceived distance gave us good indication that this data can benefit the practice of global software engineering. In most of the projects evaluated, project managers were not expecting the results found. We present the model, details of its applications, analysis of the results, lessons learned and practical implications for the management of distributed software projects and teams. Copyright © 2010 John Wiley & Sons, Ltd.

Received 17 August 2009; Revised 6 January 2010; Accepted 28 April 2010

KEY WORDS: gobal software engineering; perceived distance; distributed project management

1. INTRODUCTION

Geographically distributed work environments have became more common nowadays. Thanks to the developments in telecommunications, collaboration tools and globalization. Following this tendency, Global Software Engineering (GSE) is a promising area due to the increasing number of distributed software development (DSD) projects developed worldwide [1–5]. As teams become globally distributed, they have to face several challenges related to objective (physical) and subjective (psychological, cultural differences, etc.) distances [6–8]. Herbsleb and Moitra [8] point out that distance, time-zone and culture have diverse effects on knowledge management, as well as on strategic, cultural, and technical issues. From the perspective of the management of distributed software teams, Carmel [6] suggests the existence of five centrifugal forces (mostly related to subjective factors) that propels things outwards from the center. These forces increase the already existing distance among team members and have to be well managed to influence the success of a GSE project. O'Leary and Cummings [9] say that geographically dispersed teams are rarely 100% dispersed. For this reason, they developed a different view of geographic dispersion in teams, focusing on spatial (physical distance), temporal (time-zone) and configurational (how the team is arranged) distances. But to the best of our knowledge, no research has explored ways of making

*Correspondence to: Rafael Prikladnicki, Computer Science School, PUCRS, Porto Alegre, Rio Grande do Sul, Brazil.
†E-mail: rafaelp@pucrs.br

this subjective distance among members of distributed software teams, also known as perceived distance, more visible.

Both Evaristo *et al.* [10] and Gumm [11] suggested the concept of perceived distance to understand one person's perception of how close or how distant another person or group is. More recently, Wilson *et al.* [12] have explored the concept of perceived proximity. The motivation for their study was the lack of knowledge about what dispersion from others means from a subjective perspective, and what is the impact of distance on the performance of a globally distributed team.

Perceived distance across distributed teams is then a subjective feeling related to the perception of proximity or lack of it [10, 12]. Often, distributed teams that develop trust and mature working relationships may not feel distant from each other [12]. On the other hand, high levels of physical proximity do not necessarily lead to feelings of closeness. Related to the paradox of perceived proximity, perceived distance explains the notions of 'being far, but feeling close' and 'being close, but feeling far'. The perception of proximity becomes important because perceived distance can often be found in teams that are geographically co-located, but have difficulty in knowing what other members are working on, whereas successful distributed teams may feel very connected because of low perceived distance. Therefore, it is limiting to assume simply that geographical distribution always implies perceived distance in global teams.

In this paper, we explore the subjective dimension of propinquity, by examining the phenomenon of feeling distant (perceived distance) from geographically distributed people, in the context of distributed software teams [13–15]. This discussion is motivated not only by previous studies [10, 11], but also by situations experienced by real-world team members [16–18] where the perceived distance can be as relevant as all other types of distances. We intend to address the following research question: How perceived distance can be assessed and visible to members of GSE teams?

Our goal is to explore the concept of perceived distance in the context of GSE by presenting a model to assess and make more visible the construct of 'perceived distance' among members of GSE teams during project lifecycle. This model is called Perceived Distance Index (PDI), and it is based on the definition of perceived distance [10, 11], and perceived proximity [12]. The first instance of this model was proposed using the five centrifugal forces proposed by Carmel [6], because these forces can make distributed software work more difficult, and consequently have to be managed to improve the performance of any distributed team.

In the next section we present the theoretical background of this paper. In Section 3 we present the PDI model, followed by its usage in three globally distributed software projects. In Section 5 we present limitations, implications for research and practice, improvements opportunities and lessons we have learned. Section 6 concludes the paper.

2. GLOBAL SOFTWARE ENGINEERING

DSD can be defined as the development of software projects by a group of individuals who work across time, space and organizational boundaries, taking advantage of communication and collaboration technology [6, 8, 10]. People are distributed across multiple locations and work on the same project or product. When the distance becomes global, this characterizes the Global Software Development (GSD) [8], or, more recently, GSE [1]. The many factors that contributed to distributed development are well documented in the literature [5–8, 19, 20]. The reasons might include not only cost advantages, but also the availability of resources in different locations, proximity to local markets, quality of the work, creativity, etc. [8, 21–24].

The fact is that software development with geographically distributed projects teams is here to stay, and we have to synchronize research and practice to respond to this growing trend. Engineers, managers and executives are facing many challenges on many levels, from the technical to the social, political and cultural. People actively collaborate to achieve a common goal. And this change is having an impact on the way products are conceived, designed, tested and delivered to customers [7, 8]. This is also impacting the way projects and people are being managed [25].

DOI: 10.1002/smr

2.1. The management of distributed teams

A manager's most important and most difficult job is to manage people. In a distributed team, this task is even harder, due to the existence of physical distance. But the physical dispersion is not the only factor that influences the complexity of this job.

Kiel [26], for example, reported from a case study conducted within a software development organization where he identified five main themes based on the interviews conducted: time, language, power, culture and trust. As the author argues, none of them is particularly surprising, but the cumulative impact of the five themes is remarkable. Following Kiel, other studies have identified similar themes that characterize the complexity of managing distributed projects.

Carmel [6] has introduced two views of global software teams. The authors see software globalization as a set of a centrifugal force that propels things outwards from the center. A centrifugal force must be balanced by centripetal force, a counter force that is directed into the center. The five centrifugal forces, the problems, pull the global software team apart and inhibit its performance. The five centrifugal forces are geographic dispersion, loss of communication richness, coordination breakdown, loss of 'teamness', and cultural differences.

As centripetal forces, there is telecommunication infrastructure, collaborative technology, development methodology, product architecture, team building, and managerial techniques. In this paper, we have considered forces that may disperse the teams (centrifugal forces).

In another study, Evaristo *et al.* [10] suggest dimensions to the concept of 'distributedness'. The purpose of the study is to understand what 'distributed' means when discussing the management of distributed projects and to suggest better ways to manage distributed teams by finding out what the critical problems in such projects are. One of the dimensions proposed by the authors is related to the perception of distance among project team members, which is examined more carefully in Section 2.3.

2.2. Dimensions of dispersion in teams

There are several dimensions of team's dispersion documented in the literature. Carmel [6] suggests that distributed software teams are distant because of the geographic, temporal and cultural dispersion. O'Leary and Cummings [9] suggest that distance is related to spatial (physical), temporal and configurational dispersion. In addition, Hinds and Bailey [27] identify that physical distance can result in social and psychological effects within a team, resulting in conflicts generated by different perspectives, inconsistent norms, etc.

Although important, all dimensions were always considered in a very simple way, with a narrow view of dispersion. O'Leary and Cummings [9] were the first authors to explore a more robust way to explain dispersion, proposing five measures to better characterize the three dimensions defined by them. But all measures proposed so far have concentrated on the physical dimension of dispersion. More recently, and on understanding that dispersion can also be characterized in a more subjective way, studies have proposed a psychological dimension for dispersion, called perceived distance, or perceived proximity [10–12].

2.3. Perceived distance

In a distributed project, distance does not necessarily mean physical distance, or the physical proximity among members in each of the locations that comprise a particular project [11]. The physical distance is defined by Wilson *et al.* [12] as the objective proximity between team members.

The subjective feeling related to perception of proximity is not recently taken into consideration. Conventional wisdom makes us to expect to feel close to others who are in close proximity to us. As geographically distributed environments become more common, people might also consider the perception of proximity (or distance), which means that sometimes people will feel close to others who are miles away.

From a conceptual perspective, perceived distance across distributed sites is a subjective feeling related to the perception of proximity [10, 11]. Wilson *et al.* [12] state that members of teams with

DOI: 10.1002/smr

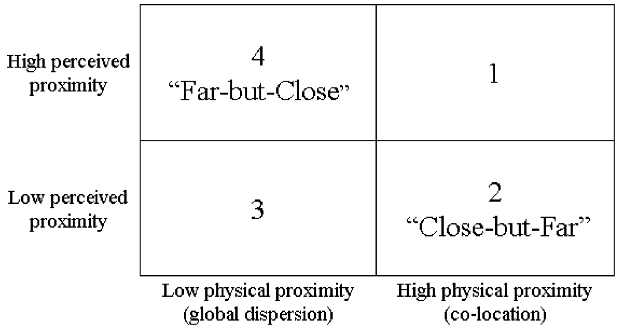


Figure 1. The paradox of perceived proximity [12].

low levels of physical proximity do not necessarily feel distant from each other. On the other hand, high levels of physical proximity do not necessarily lead to feelings of closeness. The authors call it the paradox of perceived proximity (Figure 1), explaining the notions of ‘being far but feeling close’ and ‘being close but feeling far’.

According to the authors, the perceptions of proximity (from each other or from the organization) are becoming more important, and challenges related to perceived distance might be the difficulty of not knowing who the distributed members are or what they are working on, and the lack of perception of what it means to be dispersed from others.

The paradox of ‘Close-but-Far’ is when team members are collocated, but perceived each other as distant. One example studied by the authors is the treasury analysts of banking teams. They rarely communicate with their team members even in a collocated environment. On the other hand, open-source software development project team members illustrate the paradox of ‘Far-but-Close’. Even being far from each other, they usually perceive high proximity to achieve coordination and collaboration. Based on these concepts, the authors have proposed a theoretical model of perceived proximity.

In their definition, perceived proximity is a dyadic and an asymmetric construct. It is dyadic because people form perceptions of others, and asymmetric because one can perceive distance whereas others do not have to have the same perception. The perception of proximity (or distance) refers to one’s thoughts and feelings related to others where they are involved in an interdependent task and with shared goals. In this model, individual’s perceived proximity to others is the product of their communication and identification processes, and the individual and socio-organizational factor affecting them.

3. THE PERCEIVED DISTANCE INDEX (PDI)

Motivated by previous results from field studies and the lack of studies that actually propose a way of presenting data related to this dimension of dispersion, we propose a perceived distance measure, also called PDI. Initially, the five centrifugal proposed by Carmel [6] were mapped to six factors (Table I):

The first centrifugal force is geographic dispersion, or the physical proximity among project team members. Wilson *et al.* [12] define level of dispersion as the objective physical distance between team members. Because global geographic dispersion often involves differences in time-zone, this centrifugal force was divided into physical proximity and time-zone. One may ask why physical proximity is part of the perceived distance measurement, and the reason is because we want to compare the perception of physical distance with other factors in the model. In this case, if we have situations where geographic dispersion is high but people perceive it as low, then it would make more sense having low perceptions of distance related to the other factors.

DOI: 10.1002/smr

Table I. Factors in the perceived distance index.

Centrifugal forces [6]	PDI factors
Geographic dispersion	Physical proximity and time-zone
Cultural differences	Culture
Loss of communication richness	Communication
Loss of 'teamness'	Trust
Coordination breakdown	Context sharing

The second centrifugal force is cultural differences. Hofstede [28, p. 89] defines culture as a 'collective programming of the mind which distinguishes one group or category of people from another.' The category of people in some cases can be the nation; in other cases, the organization or the individual. With distributed teams, according to Carmel and Tjia [21, p. 175], 'every adult is a member of many cultures.' This force was then mapped a factor called culture.

The last three forces build on the problem of distance: loss of communication richness, loss of 'teamness' and coordination breakdown. Loss of communication richness was mapped to communication. Loss of 'teamness' is related to how people like each other, trust each other, help each other and work harder for each other [21]. For this reason, this force was mapped to trust, since trust is seen as a moral duty, where a social group holds values its obligations to others [29]. In addition, increases in trust decrease transaction costs of relationships because people have to engage less in trust acquisition activities [29], which in the end will improve the sense of team in a group of people. The last force, coordination, is related to how well people coordinate to develop project activities [1]. Coordination breakdown is often caused by the lack of awareness of what is happening or what should happen in the course of a project [1]. Awareness and 'shared knowledge' are vital elements of coordination. For this reason, this force was mapped to context sharing. The PDI model was then proposed in three phases: Data collection, Index calculation and Analysis and action plan.

3.1. Phase 1: Data collection

In the *Data collection* phase, each project team member must answer a set of questions organized in two groups (Appendix A). The first group of questions is related to one's profile (demographic data), and it is used to calculate the individual's weight. The second group of questions is related to the six factors presented in Table I. For each factor, a question is asked, and each project team member should give his opinion using a seven-point *likert* scale.

In the future, other factors, questions, or more options can be added to the set of questions. This depends on the context of each organization, but we do not recommend working with different set of questions within the same organization or project, because this can make the comparison of the results in the future difficult. The purpose of the questionnaire is to collect data based on the individual's perception related to the project.

3.2. Phase 2: Index calculation

In the *Index calculation* phase, three indexes are calculated based on the answers received: factor index, individual index and project index. The factor index indicates how a given factor influences the team's perceived distance. The individual index represents one's perceived distance within the project, based on the factor analyzed. The project index represents the perceived distance within the project to allow comparison with other projects. The detailed calculation of each index can be found in Appendix B.

3.3. Phase 3: Analysis and action plan

We recommend the usage of the model in different points within a project (longitudinal), in order to observe the evolution (positive or negative) of the perceived distance. In this phase, the data is

DOI: 10.1002/smr

analyzed and actions are planned. One can observe the behavior of each factor and each project team member, correlating with possible cause of problems.

In some cases, the results may not have a meaningful value in the beginning. But repetition and comparisons can show interesting data such as who perceives the highest distance, or which factor is contributing the most to the perceived distance. Moreover, the analysis can be improved based on groups of data.

It is important to reach a common understanding of the meaning of each number, each index, compare with previous evaluations (if possible), and plan objective actions to improve the management of distributed software teams.

4. THE PDI IN PRACTICE

We have applied the PDI in three globally DSD projects within two different companies. Details are presented in the next sections.

4.1. Data collection and analysis

Data was collected from May through December of 2008 and was possible due to a previous contact with representatives from both companies, which explicitly manifested interest in understanding the benefits of this model for their project management activities. All respondents have at least 2 years of experience working with distributed projects. Company 1 is a global services provider located in India, with several offices around the world. A questionnaire was sent by e-mail to a representative within the company, responsible for data collection. Company 2 is a global computer company with headquarters in the US and software development centers in several countries, including Brazil, and India.

All projects were related to new development in specific areas within each company, and were following a traditional lifecycle (with an initial phase called inception followed by iterations of requirements specification, construction and transition). In addition, it is important to mention that teams were not applying agile methodologies such as Scrum or Extreme Programming. In all projects, we collected data in the middle of project execution (after the first iteration and after all project participants had the chance to interact with their colleagues). As for the Project 1, the team was working together for one year by the time the data was collected, whereas in Projects 2 and 3 teams were working together for 2 years. We made a questionnaire available online to be filled by the respondents. In the future data collection will be automated with a web-based tool. The questionnaire was the same for all participants and all project team members have answered the questions (26 people). For the project managers (PM), we also asked an additional question related to what they were expecting as the highest perception of distance within their projects and why. Table II presents the demographic data collected from each project.

As for the roles, the IT Manager is responsible for the high-level project planning and control, whereas PM are responsible for the daily project planning and monitoring. They are in charge of schedules, meetings, and all the communication flow within the project. Technical Leaders and Test Leaders are responsible for the management of all technical activities related to development and test, making sure that the team is able to execute the activities properly. Testers and Developers are responsible for the coding and testing. The Support Analysts in Project 1 are also responsible for testing, and the Environment Coordinator is responsible for making sure that every piece of code is integrated and ready for test.

4.2. Findings

After calculating all indexes (detailed data of each project can be found in Appendices C–E), we identified which project had the highest index of perceived distance, who perceived the highest and the lowest distances at that point (Figure 2) and which factors contributed to such distances

DOI: 10.1002/smr

Table II. Demographic data.

Role	Country	Years of experience with sw development	# of distributed software projects	Knowledge about distributed sw projects	Weight	Evaluation
PROJECT 1—COMPANY 1						
IT Manager	Brazil	13	10	High	9.81	Highest weight
Project Manager	Brazil	8	3	Average	5.52	Highest perceived distance
Technical Leader	India	7	10	Excellent	9.67	
Developer1	India	1.5	3	High	5.29	
Developer2	Brazil	18	1	None	4.76	Lowest perceived distance
Developer3	India	3.5	1	Low	3.00	
Support analyst1	Brazil	1	2	Low	2.86	Lowest weight
Support analyst2	Brazil	2	3	Low	3.38	
PROJECT 2—COMPANY 2						
Technical Leader1	U.S.	10	3	Average	4.26	
Project Manager	U.S.	30	50	Excellent	12.35	Highest weight Lowest perceived distance
Technical Leader2	Brazil	9	5	High	5.33	
Tester1	Brazil	12	6	High	5.72	
Test Leader	Brazil	10	20	High	6.74	
Tester2	Brazil	5	13	Low	3.63	Lowest weight Highest perceived distance
Environment coord.	Brazil	20	10	High	6.87	
Developer	Brazil	10	15	High	6.30	
PROJECT 3—COMPANY 2						
Technical Leader1	Brazil	12	14	High	8.13	
Developer1	Brazil	8	3	Average	4.49	
Developer2	Brazil	8	2	High	5.29	
Developer3	Brazil	6	4	Average	4.47	
Developer4	Brazil	20	10	High	8.22	
Environment coord.	Brazil	12	4	Average	5.13	Highest perceived distance
Technical Leader2	India	11	7	Average	5.62	
Developer5	India	5	5	Average	4.56	
Developer6	India	3.5	5	Average	4.39	Lowest weight Lowest perceived distance
Project Manager	U.S.	10	20	Average	8.11	Highest weight

(Figure 3). To double check the findings, we contacted some of the project team members and executed informal interview sections. These interviews were planned as a follow-up activity, and were executed by telephone, instant messaging, and face-to-face conversations.

As for the project index, we found 43.75% for Project 1, 41.67% for Project 2 and 43.57% for Project 3. This index alone was not enough to explain the perception of distance within the projects. For this reason, we investigated the other indexes, and interesting results were found.

DOI: 10.1002/smr

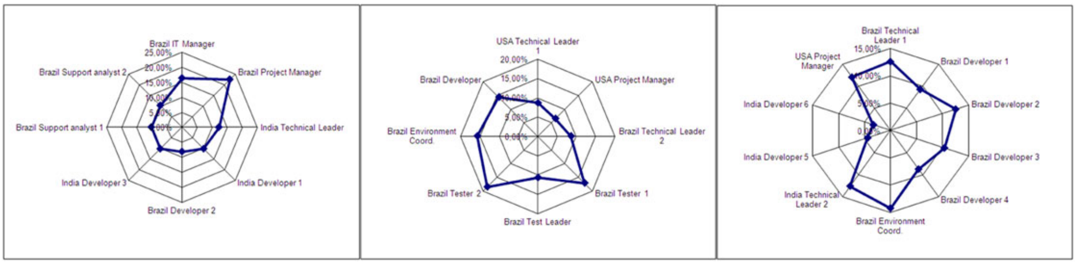


Figure 2. Perceived distance by team members in projects 1, 2 and 3 respectively.

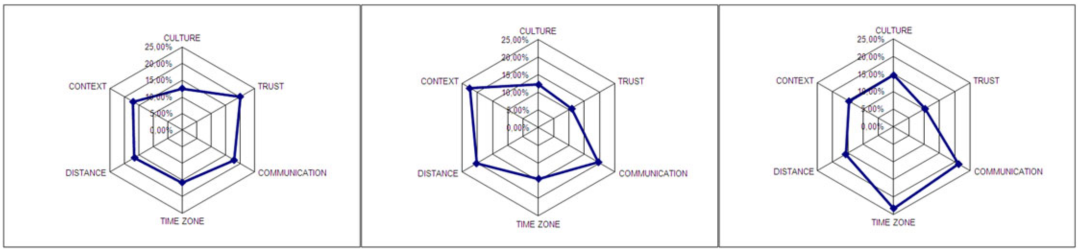


Figure 3. Perceived distance by factor in projects 1, 2 and 3 respectively.

Table III. Most critical factors in the perceived distance.

	Project 1	Project 2	Project 3
Factor	Communication	Context sharing	Time-zone

Before presenting data related to the perceived distance, Table III shows what each PM was expecting as the most critical factors in the perceived distance calculated.

For Project 1 the PM said that among the six factors evaluated, COMMUNICATION was the most critical to deal with. The main reason was the need to talk to people from India very often. Based on the manager's view, the team was also facing some problems related to the time-zone difference. In Project 2, the PM cited that the two people working from the U.S. were lacking the exchange of task-related information such as who was doing what, and he was having problems trying to solve this issue. For this reason, the team in the U.S. was not completely aware of what was going on in the project, and the most critical factor was CONTEXT SHARING. The PM also mentioned that the lack of context sharing could be impacting TRUST. In Project 3, the PM mentioned that the most difficult was to deal with the TIME-ZONE difference. Because the team was located in Brazil, India and the U.S., time-zone was impacting team's performance and communication.

The PDI was then calculated, and Figure 2 presents who perceived the highest and the lowest distances in the three projects. We used radar charts, also known as spiderweb charts to present all the results because it shows strengths and weaknesses at a glance [21]: the larger the area represented in the chart, the worse the changes to have a well-integrated team.

One can observe different roles perceiving the highest distance in each project (a PM in Project 1, a Tester in Project 2 and the Environment Coordinator in Project 3). In addition, it seems that Project 1 is the one with the team better integrated. On the other hand, Project 2 has half of the team perceiving high levels of distance, and in Project 3 most of the team perceives high levels of distance.

In Figure 3, one can also observe different factors contributing to the perceived distance in each project. While in Project 1 the main factor was TRUST, in Project 2 was CONTEXT SHARING and in Project 3 was TIME-ZONE.

DOI: 10.1002/smr

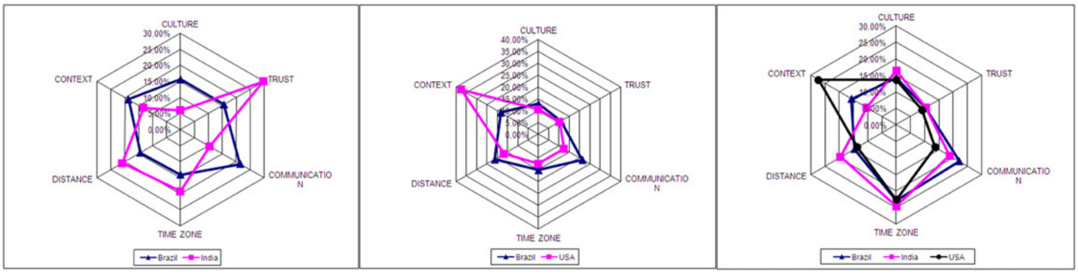


Figure 4. Data grouped by country in projects 1, 2 and 3 respectively.

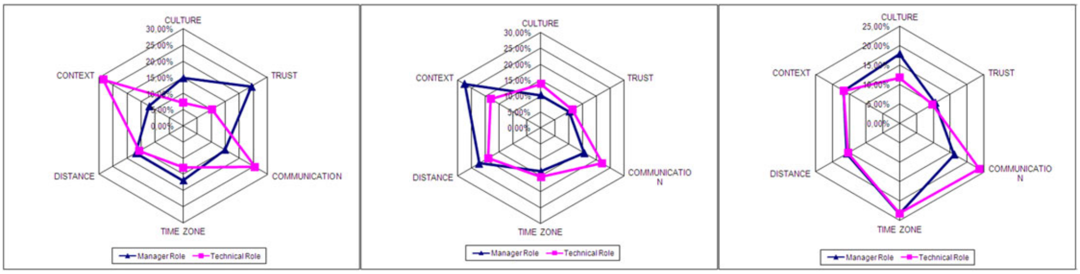


Figure 5. Data grouped by role in projects 1, 2 and 3 respectively.

Based on this first set of data, the obvious conclusion is that managers and teams should be flexible to understand and deal with different human factors among different projects, and that each project is different. The interesting finding is that the results reflected exactly what the project participants were feeling in terms of perceived distance, but not exactly what the PM were expecting. In Project 1, for example, TRUST was ranked as the factor with the highest perception of distance, whereas the PM answered COMMUNICATION. In this case, the PM would have not solved one of the main team's issues if he would have planned actions based on his perception. A possible explanation for this behavior is illustrated later in the paper (Figure 4), where TRUST was the most critical factor based on the perception of the Indian team members and COMMUNICATION was based on the perception of the Brazilian team members. In other words, the PM was influenced by his local team. In the other two projects, PM were aligned with the team, despite the fact that in Project 2 the lack of CONTEXT SHARING was not impacting TRUST as suggested by the PM.

The perception of proximity also made it possible to identify intriguing situations. As examples, in Projects 2 and 3, we had two participants (Tester2 and Developer6, respectively) with the lowest weight, indicating less experience than others (Table CII, last column). While the first perceived the highest distance, the second perceived the lowest distance within the project environment. This finding is particularly interesting because one can argue that the weighting scheme means that the problems of the less experienced developers may appear to be less important. But this may not be the case, as we illustrated in this situation.

Another analysis was done by grouping the data by country (Figure 4) and role (Figure 5). In Project 1 the predominant factor in the perceived distance in India was TRUST, while in Brazil it was COMMUNICATION. In Project 2, CONTEXT SHARING was the predominant factor in the US, while in Brazil it was COMMUNICATION. In Project 3, the perceptions in Brazil and India were pretty much the same, and TIME-ZONE was the predominant factor. In the US the predominant factor was CONTEXT SHARING. All findings were confirmed by the project team members during informal conversations. As an example, in Projects 2 and 3 both teams identified that TRUST was not contributing to the perceived distance because the teams were working together for at least 2 years, and had the chance to improve trust over the years.

DOI: 10.1002/smr

Table IV. Summary of findings.

	Project 1	Project 2	Project 3
General	Trust	Context sharing	Time-zone
Role: Managerial	Trust	Context sharing	Time-zone
Role: Technical	Context sharing	Communication	Time-zone and Communication
Country: Brazil	Communication	Communication and Distance	Time-zone
Country: India and/or U.S.	Trust (IN)	Context sharing (U.S.)	Time-zone (IN) and Context sharing (U.S.)

Regarding roles, three were grouped as managerial (IT Manager, PM and Technical Lead), and the remaining were grouped as technical.

In Project 1, most of the technical people were lacking CONTEXT SHARING, and most of the managerial people were having problems trusting (TRUST) their peers. This was also corroborated during informal conversations, and was quite a surprise in the team's view, mainly because the PM was sharing the same perception of the Brazilian team, as explained before. With this analysis, the team was able to identify weaknesses that were not visible before, and plan actions based on different expectations within the project. In Project 2, most of the technical people were facing COMMUNICATION issues, while the managerial team was lacking CONTEXT SHARING. In Project 3, both technical and managerial people were facing TIME-ZONE differences, but technical people were also facing COMMUNICATION problems. In general, all participants found the model useful because it was possible to identify important issues related to the factors analyzed before the factor becomes a problem itself, and with the advantage of identifying the factors during the project lifecycle. Table IV summarizes all findings.

On analyzing this table, one can observe that only in Project 3 a common factor was contributing to the perceived distance from the point of view of all participants, roles and countries. As for the Projects 1 and 2, general perception was not followed completely when data was grouped. As a result, the teams had the opportunity to plan different activities to foster a better integration depending on the country and the role. In Project 1, for example, the team in Brazil suggested ways to improve communication, while the technical team suggested ways to better coordinate the work and made the progress visible to everybody, leveraging TRUST within the team.

5. DISCUSSION

The quantification of perceived distance has as purpose to serve as an additional input to help PM in the management of distributed software projects and teams. On proposing this model, we have established a first attempt to provide numbers to represent subjective feelings, and interpret these numbers within the project environment. In addition, an important contribution of this work is the possibility of using this model during the project lifecycle, instead of doing post mortem analyses. In this section, we discuss the implications of this model for research and practice. We also discuss limitations already identified, and improvement opportunities.

5.1. Implications for practice

DSD is here to stay. Costs reduction, access to skilled resources anywhere and several other advantages make DSD a necessity. But a distributed software project team should be well managed. Managers of distributed teams (globally or not) face several challenges related to human factors. On providing a way to quantify the perceptions of each project team member, one can try to be more proactive, making subjective things more visible.

DOI: 10.1002/smr

Wilson *et al.* [12] also argue that 'managers do not have a good model of what influences relationships at a distance'. For this reason, most of the time they decide to bring people together face-to-face. If perceived distance starts to be visible and as a consequence properly managed, this may reduce costs of bringing team members together in every project.

Another implication for the practice of distributed and GSD is that, based on the quantification of perceived distance, people will start to identify patterns for certain teams, clusters of teams, individuals and this may help in the selection of people to be part of certain projects, or anticipate risks. If a global team is well composed, physical contact may not be as necessary as we see nowadays. Moreover, this work can impact training and organization development initiatives, since the perception of distance will provide evidence of aspects that are missing within a specific company, subsidiary of a company or a project. As an example, certain levels of perceived distance might indicate that training is needed for people located in certain countries or performing certain roles. The model and its evaluation in three real-world cases also showed that sometimes managers of distributed projects do not have the complete view of team's perception until something not expected happens to the project. The PDI could be an approach that makes data visible in a way that managers can try to avoid something to happen.

Finally, we believe that the perception of distance (or proximity) is becoming more important than the physical distance itself. For this reason, although the quantitative evaluation of perceived distance makes more sense in the context of distributed teams, this can also be applied to local teams that want to evaluate their perceived distance.

5.2. Implications for research

There are several research possibilities regarding the usage of this model, or the quantification of subjective factors itself. One example is the correlation between perceived distance and physical distance, or between perceived distance and several variables within a certain project, such as issues, delays and other aspects of project performance. We also understand that the measurement of perceived distance can be correlated with a metric called socio-technical congruence (STC), proposed by Cataldo [30]. The author has proposed a metric to investigate the alignment between the software architecture and the communication and coordination aspects of the work. Their results suggest that high degrees of STC correlate with task performance, i.e. the more productive software developers are those who have high degrees of congruence. Our interest in this case is to study the reasons of low degrees of STC, trying to answer the following question: What is the degree of perceived distance of developers with low degrees of STC?

Another research opportunity is to combine perceived distance with recent results of case studies that reported successful execution of distributed projects using the agile method known as Scrum as the main project management strategy [31]. Based on interviews with 19 project team members, the authors have described how Scrum practices were successfully adopted to distributed development, and concluded that Scrum can help GSE because it better introduces communication since the beginning. For this reason, research may be developed in order to compare distributed teams that use Scrum or traditional project management strategies and evaluated the levels of perceived distance in each team. One may investigate to what extent Scrum accelerates the integration and coordination of distributed teams, by analyzing the perception of distance within the project. These results will also impact the practice of GSE.

Another suggestion is to add to the work of O'Leary and Cummings [9] and propose an additional measure—focusing on the psychological dimension of dispersion—to improve the characterization of the existing dimensions proposed by them. In addition, the usage of this model can lead people to identify interesting patterns of perceived distance within teams. As an example, should we have a team that perceives less distance in a project where tasks do not demand strong collaboration? Or what is the maximal level of perceived distance for a team to operate successfully? These and other questions can be a subject of future investigations.

5.3. Limitations

The PDI model was proposed having DSD project teams as the focus and cannot be generalized this time. In the future, the model may be adapted to be used by other project team members (not only distributed, and not only software teams). The evaluation of this model is limited to the perceived distance in a single project. As significant project team members start using the model, the historical data might generate interesting analysis, including the evolution of perceived distance grouped by data such as portfolio of projects and additional roles.

Other limitations are related to the simplicity of questions being asked. Our idea in the beginning was to have simple questions, simple questionnaires and meaningful results. The numbers are less important than who perceives the highest distance, and which factors contribute to the perceived distance. The numbers play a role of quantification, but the most important is the interpretation given to them and the analysis based on the context of each project. The questions then are useful to calculate the indexes, but they do not explain the reasons of the existing distance. We recommend follow-up interviews to understand the context of each project.

Additional limitation is related to how the evaluation is replicated within the same project. We should take into consideration the turn over factor, and that people may be fired or may be hired with the project running. This may be a confounding factor and must be known as soon as possible, avoiding discrepancies in the analysis.

5.4. Improvements opportunities

Besides the evaluation of the PDI model in three real-world cases, we also evaluated the model with experts in the field, collecting feedback and improvement opportunities. The model was presented to a DSD practitioner, a DSD researcher and a group of 20 practitioners from a DSD company. In all cases, the model was presented, using the same material, which were a presentation and a spreadsheet with simulated data. Table V presents suggestions that were received and are being evaluated for the model improvement.

Each person was invited to give feedback without knowing other's feedback. All suggestions were evaluated together with other initiatives that have been planned, such as data collection and data analysis automation. The automation is intended to facilitate comparisons and an easier access to historical data. To this end, we are developing a web-based tool to better support the usage of this model in several projects, making this tool also available to the community. The identification of comparison attributes means that in the future projects could be compared based on their characteristics and the perceived distance found. For example, after several rounds of data collection from a significant number of projects within the same company, one may find that for new projects that have teams working together for a period of less than 12 months and part of the team is located in Brazil, context sharing is most likely to be an issue after the first iteration. The professionals from Psychology could contribute to the improvement of the model on adding, for example, more human factors to be evaluated. The last two suggestions are based on the model

Table V. Improvements identified.

Improvement item	DSD Practitioner	DSD Researcher	DSD Company
Automate data collection with information from projects (answers may be manipulated)	X		
Identify comparison attributes for different projects	X		
Involve professionals from Psychology in an interdisciplinary project		X	X
Take into consideration if it is a newly formed team, or if people have worked together		X	
Take into account multiple roles that a team member might play		X	

DOI: 10.1002/smr

itself and the characteristics of the teams being evaluated. First, one should take into account if the team is new or have been working together for a long time. To this end, historical data of team allocation should be collected and reflected into the model. And second, one should consider that an individual might play multiple roles within the same project. In this case, we have to decide if we consider all roles or only the most significant role for each person.

5.5. Lessons learned

By quantifying perceived distance (as one of the propinquity dimensions) we identified that this data can benefit the practice of GSE and create additional questions for further research. In most of the projects evaluated, PM were not expecting the results found and they planned actions just after the analysis was presented. In addition to that, we have identified four lessons learned:

#1: *The quantitative management of human factors within distributed teams can make things more visible*: The PDI model is a management technique that was initially proposed having human factors as the factors to be quantitatively evaluated. Informal conversations with managers and other respondents have confirmed that the data provided by the model can really make invisible and subjective things more visible, and as consequence can benefit the management of distributed teams.

#2: *The quantification of perception can help in the management of distributed software teams*: The concept of perceived distance is not new. Project teams understand that distance is not only physical. But studies exploring this concept in the context of GSE are new. Our experience has shown that the quantification of perception may benefit managers of distributed software teams.

#3: *The quantification of perceived distance may be relevant not only in the context of distributed software projects*: The PDI model was initially proposed to be used in the context of distributed software teams. We argue that this model may be used to quantify distance even if the distance appears to be insignificant. Once the perceived distance is quantified, interesting results may arise and change some beliefs that collocated teams always feel close.

#4: *The PDI model can be configured to work with other factors*: The PDI model was initially proposed with six factors based on the five centrifugal forces proposed by Carmel [6]. We have learned that other factors can be added, and the results will have the same effect in terms of quantifying perceived distance. The difference will be in the analysis. To have this flexibility, one should follow the rules and formulas defined.

6. CONCLUSIONS

Projects are executed by people. Distributed projects are not different. Many times, the management of distributed software teams is done in a reactive basis. For this reason, managers are most likely to identify problems, instead of potential risks, and do not have enough time to react. Such problems are frequently related to human factors, and the management of human factors is a critical success factor for DSD teams [11, 26]. Perceived distance is an important aspect that has to be considered in this context, because it adds another dimension to the understanding of dispersion in teams: the psychological dimension.

In this paper, we presented the PDI model, a way to make perceived distance more visible to software PM. This approach can help to minimize the occurrence of problems—mainly related to human factors—within a distributed project, making potential risks visible to managers early in the project, and during the project execution. On proposing this model we have established an attempt to provide numbers to represent subjective feelings, and interpret these numbers within the project environment, serving as an additional input to help PM in the management of distributed software projects and teams.

DOI: 10.1002/smr

APPENDIX A: QUESTIONNAIRE

Demographic data

1. Role within the project (according to the company's terminology). If more than one please specifies as many roles you have:
2. Country from where you work:
3. Country where you were born:
4. Years of experience with software development
5. Number of projects you were involved where the project had **distributed teams** (if only the client was distributed, please don't consider as a project with distributed teams)?
6. Knowledge about Global Software Engineering
   1. ☐ None 2. ☐ Low 3. ☐ Average 4. ☐ High 5. ☐ Excellent

Questions related to Factors

Specific questions: these questions are related to the project specifically. Please consider only the project environment, and select between 1 (lowest impact) and 7 (highest impact) for each factor.

1. Communication: Do you have communication problems in your project (related to misunderstandings, difficulties on reaching people, etc.)?

1	2	3	4	5	6	7

2. Physical distance: Is the physical distance a problem in your project?

1	2	3	4	5	6	7

3. Time-zone: Do time-zone differences affect negatively your project?

1	2	3	4	5	6	7

4. Culture: Do cultural differences (national differences, individual differences) affect negatively your project?

1	2	3	4	5	6	7

5. Context sharing: Do you have difficulties in knowing what your project teams members are doing?

1	2	3	4	5	6	7

6. Trust: Do you have difficulties on trust your project team members?

1	2	3	4	5	6	7

APPENDIX B: HOW TO CALCULATE THE THREE INDEXES

To calculate the indexes, we have assumed that:

- i represents a given factor;
- j represents a certain person (also a respondent);
- n represents the number of people being evaluated;
- x represent the total number of factors (six);
- z represents the max value in a *likert* scale (seven);
- $W(j)$ represents the weight of a certain person j;

The weight is used to adjust the values, and it is calculated based on the first group of questions (demographic data). The weight was calculated following the weighting scheme proposed by Dias

DOI: 10.1002/smr

Neto *et al.* [32], and it is described as following:

$$W(j) = \frac{E(j)}{MedianE} + \frac{QDP(j)}{MedianQDP} + K(j),$$

where:

- $E(j)$ is the experience, in years, of a person j in software development;
- *MedianE* is the median of experiences;
- *QDP(j)* is the total number of distributed software projects in which a person j has been involved so far;
- *MedianQDP* is the median of the number of distributed software projects;
- $K(j)$ is the level of knowledge of a person j in DSD.

Factor index. To calculate this index, we multiply the answer of each respondent by its weight: $AdjFactor(i)(j) = Answer(i)(j) * W(j)$, where:

- AdjFactor(i)(j) is the adjusted number of a factor i for a person j;
- Answer(i)(j) is the answer given by a person j to a factor i;

The next step is to calculate the total for each factor, as following:

$$TotalFactor(i) = \sum_{j=1}^{j=n} AdjFactor(i)(j),$$

where:

- TotalFactor(i) is the total of the factor i for all people;
- AdjFactor(i)(j) is the adjusted number of a factor i for a person j;

The total is then normalized, divided by the maximum possible value, as following:

$$NormFactor(i) = \frac{TotalFactor(i)}{z * \left(\sum_{j=1}^{j=n} W(j)\right)},$$

where:

- NormFactor(i) is a percentage number representing a normalized value for a factor i;
- TotalFactor(i) is the number calculated in the previous step;

Last, but not least, we calculate the percentage for each factor in a range between 0 and 100:

$$IndexFactor(i) = \frac{NormFactor(i)}{\sum_{i=1}^{i=x} NormFactor(i)},$$

where:

- IndexFactor(i) is the PDI for a factor i;
- NormFactor(i) is the number calculated before;

Individual index. To calculate this index, we use the adjusted number calculated previously, and generate the total for each respondent, as following:

$$TotalPerson(j) = \sum_{i=1}^{i=x} AdjFactor(j)(i),$$

where:

- TotalPerson(j) is the total of the person j for all factors;
- AdjFactor(j)(i) is the adjusted number of a person j for a factor i;

DOI: 10.1002/smr

The total is then normalized, divided by the maximum possible value, as following:

$$NormPerson(j) = \frac{TotalPerson(j)}{W(j)*x*z},$$

where:

- NormPerson(j) is a percentage number representing a normalized value for a person j;
- TotalPerson(j) is the number calculated in the previous step;

We then calculate the percentage for each person:

$$IndexIndividual(j) = \frac{NormPerson(j)}{\sum_{j=1}^{j=n} NormPerson(j)},$$

where:

- IndexIndividual(j) is the PDI for a person j;
- NormPerson(j) is the number calculated before;

Project index. To calculate this index, we use the average of the normalized values found for each project team member, as following:

$$IndexProject = \frac{\sum_{j=1}^{j=n} NormPerson(j)}{n},$$

where:

- IndexProject is the PDI for a project being evaluated;
- NormPerson(j) is the number calculated before;

APPENDIX C: PROJECT 1

The detailed data of Project 1 is given in Tables CI–CIII.

Table CI. Indexes calculation.

Country	Role	Culture	Trust	Communication	Time-zone	Distance	Context	Weight	Normalized (%)	Person index (%)
Brazil	IT Manager	39.24	39.24	39.24	39.24	39.24	39.24	9.81	57.14	16.33
Brazil	Project Manager	38.67	38.67	38.67	22.10	22.10	22.10	5.52	78.57	22.45
India	Technical Leader	9.67	67.67	9.67	38.67	38.67	9.67	9.67	42.86	12.24
India	Developer1	5.29	5.29	21.14	5.29	21.14	21.14	5.29	35.71	10.20
Brazil	Developer2	4.76	4.76	19.05	4.76	4.76	19.05	4.76	28.57	8.16
India	Developer3	3.00	12.00	3.00	12.00	3.00	12.00	3.00	35.71	10.20
Brazil	Support analyst1	2.86	2.86	11.43	11.43	2.86	11.43	2.86	35.71	10.20
Brazil	Support analyst2	3.38	3.38	13.52	3.38	13.52	13.52	3.38	35.71	10.20
Total		106.86	173.86	155.71	136.86	145.29	148.14	44.29		
Normalized (%)		34.47	56.08	50.23	44.15	46.87	47.79			
Factor index (%)		12.33	20.06	17.97	15.79	16.76	17.09			

DOI: 10.1002/smr

Table CII. Data grouped by country.

Country	Culture (%)	Trust (%)	Communication (%)	Time-zone (%)	Distance (%)	Context (%)
Brazil	15.64	15.64	21.45	14.23	14.51	18.53
India	6.02	28.48	11.33	18.76	21.06	14.35

Table CIII. Data grouped by role.

Role	Culture	Trust	Communication	Time-zone	Distance	Context
Managerial	14.80	24.60	14.80	16.90	16.90	12.00
Technical	7.01	10.29	24.78	13.40	16.47	28.05

APPENDIX D: PROJECT 2

The detailed data of Project 2 is shown in Tables DI–DIII.

Table DI. Indexes calculation.

Country	Role	Culture	Trust	Communication	Time-zone	Distance	Context	Weight	Normalized (%)	Person index (%)
USA	Technical Leader1	4.26	4.26	8.52	8.52	17.04	8.52	4.26	28.57	8.57
USA	Project Manager	12.35	12.35	12.35	12.35	12.35	49.39	12.35	21.43	6.43
Brazil	Technical Leader2	10.67	10.67	10.67	10.67	10.67	10.67	5.33	28.57	8.57
Brazil	Tester1	22.89	11.44	28.61	34.33	22.89	17.17	5.72	57.14	17.14
Brazil	Test Leader	6.74	6.74	20.22	13.48	33.70	20.22	6.74	35.71	10.71
Brazil	Tester2	14.52	7.26	18.15	18.15	14.52	21.78	3.63	61.90	18.57
Brazil	Environment coord.	13.74	13.74	41.22	13.74	41.22	27.48	6.87	52.38	15.71
Brazil	Developer	18.91	25.22	25.22	12.61	18.91	25.22	6.30	47.62	14.29
Total		104.08	91.68	164.95	123.85	171.30	180.44	51.21		
Normalized (%)		29.03	25.58	46.02	34.55	47.79	50.34			
Factor index (%)		12.45	10.96	19.72	14.81	20.48	21.58			

Table DII. Data grouped by country.

Country	Culture (%)	Trust (%)	Communication (%)	Time-zone (%)	Distance (%)	Context (%)
Brazil	12.98	11.14	21.38	15.28	21.05	18.18
India	10.24	10.24	12.86	12.86	18.11	35.69

Table DIII. Data grouped by role.

Role	Culture (%)	Trust (%)	Communication (%)	Time-zone (%)	Distance (%)	Context (%)
Managerial	10.39	10.39	15.81	13.75	22.53	27.13
Technical	13.77	11.33	22.24	15.49	19.17	18.01

APPENDIX E: PROJECT 3

The detailed data of Project 3 is shown in Tables EI–EIII.

Table EI. Indexes calculation.

Country	Role	Culture	Trust	Communi-cation	Time-zone	Distance	Context	Weight	Normal-ized (%)	Person index (%)
Brazil	Technical Leader1	40.67	24.40	32.53	40.67	24.40	24.40	8.13	54.76	12.57
Brazil	Developer1	4.49	13.47	22.44	8.98	8.98	17.96	4.49	40.48	9.29
Brazil	Developer2	15.87	5.29	21.16	31.73	26.44	21.16	5.29	54.76	12.57
Brazil	Developer3	4.47	13.40	17.87	17.87	13.40	17.87	4.47	45.24	10.38
Brazil	Developer4	16.44	8.22	41.11	32.89	16.44	16.44	8.22	38.10	8.74
Brazil	Environment Coord.	20.53	10.27	25.67	35.93	20.53	20.53	5.13	61.90	14.21
India	Technical Leader2	22.49	11.24	22.49	33.73	28.11	11.24	5.62	54.76	12.57
India	Developer5	4.56	4.56	9.11	9.11	4.56	4.56	4.56	19.05	4.37
India	Developer6	4.39	4.39	4.39	4.39	4.39	4.39	4.39	14.29	3.28
USA	Project Manager	24.33	16.22	24.33	40.56	24.33	48.67	8.11	52.38	12.02
Total		133.90	95.23	196.77	215.30	147.26	138.54	50.30		
Normalized (%)		38.03	27.05	55.88	61.15	41.82	39.35			
Factor index (%)		14.44	10.27	21.23	23.23	15.89	14.95			

Table EII. Data grouped by country.

Country	Culture (%)	Trust (%)	Communication (%)	Time-zone (%)	Distance (%)	Context (%)
Brazil	13.94	10.21	21.88	22.87	15.00	16.10
India	16.36	10.51	18.74	24.59	19.29	10.51
USA	13.64	9.09	13.64	22.73	13.64	27.27

Table EIII. Data grouped by role.

Role	Culture (%)	Trust (%)	Communication (%)	Time-zone (%)	Distance (%)	Context (%)
Managerial	17.68	10.48	16.04	23.23	15.53	17.04
Technical	11.59	9.76	23.21	23.07	15.52	16.85

ACKNOWLEDGEMENTS

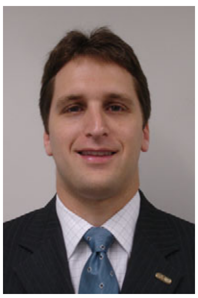
Study developed by the MuNDDoS Research Group on Distributed Software Development and partially supported by the PDTI program, financed by Dell Computers of Brazil Ltd. (Law 8.248/91). We also thank the company and all the participants who voluntarily answered the questionnaire.

REFERENCES

1. Herbsleb JD. Global software engineering: The future of socio-technical coordination. *Twenty-ninth International Conference on Software Engineering*, 2007; 188–198.
2. Sangwan R, Bass M, Mullick N, Paulish DJ, Kazmeier J. *Global Software Development Handbook*. Auerbach Publications: Boca Raton NY, 2007.
3. Boehm B. A view of 20th and 21st century software engineering. *In the 28th International Conference on Software Engineering*, 2006; 12–29.
4. Ramasubbu N, Krishnan MS, Kompalli P. Leveraging global resources: A process maturity framework for managing distributed development. *IEEE Software* 2005; **22**(3):80–86.
5. Aspray W, Mayadas F, Vardi MY. Globalization and offshoring of software. *A Report of the ACM Job Migration Task Force*, Association for Computing Machinery, 2006.
6. Carmel E. *Global Software Teams—Collaborating Across Borders and Time-Zones*. Prentice-Hall: U.S.A., 1999.
7. Damian D, Moitra D. Guest editors' introduction: Global software development: How far have we come? *IEEE Software* 2006; **23**(5):17–19.
8. Herbsleb JD, Moitra D. Guest editors' introduction. *IEEE Software* 2001; **18**(2):16–20.

9. O'Leary MB, Cummings JN. The spatial, temporal, and configurational characteristics of geographic dispersion in teams. *Management Information Systems Quarterly* 2007; **31**(3):433–452.
10. Evaristo R, Scudder R, Desouza K, Sato O. A dimensional analysis of geographically distributed project teams: A case study. *Journal of Engineering and Technology Management* 2004; **21**(3):75–189.
11. Gumm DC. Distribution dimensions in software development projects: A taxonomy. *IEEE Software* 2006; **23**(5):17–19.
12. Wilson JM, O'Leary MB, Metiu A, Jett QR. Perceived proximity in virtual work: Explaining the paradox of far-but-close. *Organization Studies* 2008; **29**(07):979–1001.
13. Bradner E, Mark G. Why distance matters: Effects on cooperation, persuasion and deception. *CSCW*, 2002; 226–235.
14. Burgoon JK, Bonito JA, Ramirez A, Dunbar NE, Kam K, Fischer J. Testing the interactivity principle: Effects of mediation, propinquity, and verbal and nonverbal modalities in interpersonal interaction. *Journal of Communication* 2002; **52**:657–677.
15. Nahemow L, Lawton MP. Similarity and propinquity in friendship formation. *Journal of Personality and Social Psychology* 1975; **32**(2):205–213.
16. Prikladnicki R, Audy JLN, Damian D, Oliveira TC. Distributed software development: Practices and challenges in different business strategies of offshoring and onshoring. *Second International Conference on Global Software Engineering*, 2007; 262–271.
17. Prikladnicki R, Audy JLN, Evaristo R. A reference model for global software development: Findings from a case study. *First International Conference on Global Software Engineering*, 2006; 18–25.
18. Prikladnicki R, Audy JLN, Evaristo JR. Global software development in practice: Lessons learned. *Journal of Software Process Improvement and Practice* 2003; **8**(4):267–281.
19. Layman L, Williams L, Damian D, Bures H. Essential communication practices for extreme programming in a global software development team. *Journal of Software and Technology* 2006; **48**(9):781–794.
20. Nguyen T, Wolf T, Damian D. Global software development and delay: Does distance still matter? *Proceedings of the International Conference on Global Software Engineering* (*ICGSE*), 2008.
21. Carmel E, Tjia P. *Offshoring Information Technology*: *Sourcing and Outsourcing to a Global Workforce*. Cambridge University Press: Cambridge, 2005.
22. Komi-Sirviö S, Tihinen M. Lessons learned by participants of distributed software development. *Knowledge and Process Management* 2005; **12**(2):108–122.
23. Meyer B. The unspoken revolution in software engineering. *IEEE Computer* 2006; **39**(1):124, 121–123.
24. Sengupta B, Chandra S, Sinha V. A research agenda for distributed software development. *In the 28th International Conference on Software Engineering*, 2006; 731–740.
25. Herbsleb JD, Paulish DJ, Bass M. Global software development in practice: Experience from nine projects. *Twenty-seventh International Conference on Software Engineering*, 2005; 524–533.
26. Kiel L. Experiences in distributed development: A case study. *International Workshop on Global Software Development at ICSE*, 2003; 44–47.
27. Hinds PJ, Bailey DE. Out of sight, out of sync: Understanding conflict in distributed teams. *Organization Science* 2003; **14**(6):615–632.
28. Hofstede G. Cultural constraints in management theories. *Academy of Management Executive* 1993; **7**(1):81–94.
29. Jarvenpaa SL, Knoll K, Leidner DE. Is anybody out there? Antecedents of trust in global virtual teams. *Journal of Management Information Systems* 1998; **14**(4):29–64.
30. Cataldo M. Dependencies in geographically distributed software development overcoming the limits of modularity. *PhD Thesis*, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 2007.
31. Paasivaara M, Durasiewicz S, Lassenius C. Using scrum in distributed agile development: A multiple case study. *Fourth International Conference on Global Software Engineering*, 2009; 195–204.
32. Dias Neto AC, Natali ACC, Rocha AR, Travassos GH. Characterization of the state of the practice of testing activities in Brazil. *The V Brazilian Symposium on Software Quality* (*SBQS*) (in Portuguese), 2006.

AUTHOR'S BIOGRAPHY



Rafael Prikladnicki is a Professor in the Computer Science School at the Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS). He is also project manager coordinator at PUCRS's Technological Management Agency. Professor Rafael's areas of expertise are distributed software development and agile methodologies. His 2007 book 'Distributed Software Development—developing software with distributed teams' was the first Portuguese book on this topic. He also leads one of the main research groups in this area in Brazil (MuNDDoS research group), and is especially interested in how global software engineering (GSE) interplay with agile methodologies, how GSE impacts organizational decisions and the role of Brazil in the global IT industry.

DOI: 10.1002/smr