Team Software Process in GSD Teams: A study of new work practices and models

Adrián Hernández-López
Computer Science Department,
Universidad Carlos III de Madrid
Av. Universidad, 30.
28911 Leganés (Madrid). Spain
Phone: +34 91 624 5958
Fax: +34 91 624 9129
adrian.hernandez@uc3m.es

Ricardo Colomo-Palacios (*)
Computer Science Department,
Universidad Carlos III de Madrid
Av. Universidad, 30.
28911 Leganés (Madrid). Spain
Phone: +34 91 624 5958
Fax: +34 91 624 9129
ricardo.colomo@uc3m.es

Ángel García-Crespo
Computer Science Department,
Universidad Carlos III de Madrid
Av. Universidad, 30.
28911 Leganés (Madrid). Spain
Phone: +34 91 624 5958
Fax: +34 91 624 9129
angel.garcia@uc3m.es

Pedro Soto-Acosta
Department of Management & Finance
University of Murcia,
Campus de Espinardo, 30100 Murcia, Spain
Phone: +34 868 887805
Fax: +34 868 887537

psoto@um.es

(*) Corresponding Author

Abstract

Distributed software development is becoming the norm for the software industry today as an organizational response to globalization and outsourcing tendencies. In this new environment, centralized models for software development team building models have to be reanalyzed. Team Software Process (TSP) guides engineering teams in developing software-intensive products and is intended to improve the levels of quality and productivity of a team's software development project. In this paper, the authors aim to assess the difficulty of using TSP in distributed software development environments. The objective of this assessment is twofold; firstly, know the general difficulty for using TSP in these environments, and secondly, know the caveats to be addressed in future software development team building models designed specifically for distributed environments.

Keywords

IT Outsourcing; Team Software Process; Software Development; Team Management; Software Development Teams.

Introduction

Software development is a collaborative and knowledge intense process where success depends on the ability to create, share and integrate information (Walz et al., 1993). This process has been evolving since its origins in the NATO Software Engineering Conference at end of the 60's. During this evolution, new forms of software development have appeared, i.e. open-source software development (Lerner & Tirole, 2002) or outsourcing development (Sahay et al., 2003). Along with the evolution of software development, the research about it has evolved and branched too. Moreover, an external but nearly related factor to software development process and its characteristics is still a top issue since, which are people (Gannon, 1979; Laughery Jr & Laughery Sr, 1985);*"People factors" seem to dominate "tools and techniques"* as Blackburn et al. (1996) pointed in their study. Within people issue, other issues such as teams (Krishnan, 1998), people's factors (Blackburn et al., 1996), and human resources management issues such as assigning roles (Acuña & Juristo, 2004), productivity management of IT projects (Mahmood et al., 1996) or skills identification (Colomo et al., 2010; Trigo el al., 2010) as top issues in software development research.

As a key issue for software development process, quality improvement within an organization is founded on three basic pillars: processes, technologies and people. These pillars are interconnected with each other, forming an inseparable triangle on which organizations operates. In turn, software process improvement requires an effort on three levels: organization, team, people (Humphrey, 2002). For each one of these levels there are well-recognized models for its continuous improvement and quality. At organizational level, models such as CMMI (Capability Maturity Model Improvement), which based on the premise that management is convinced of the need to improve processes, facilitates the integration of traditional functions, establishes objectives and priorities for processes improvement in the guide to quality processes, and evaluates current processes; at team level, models such as TSP (Team Software Process), which guides engineers in the creation and maintenance of self-managed software teams, which plan and monitor their work, set their goals and commit to compliance with the them (Humphrey, 2000); and at individual level, models such as PSP (Personal Software Process), which shows software engineers to manage the quality of their projects, agreeing that can meet commitments to improve their estimates and schedules and reduces defect rates in their products (Humphrey, 1995).

A software development team is defines as a team that has the implementation of a software project as its mission (Sawyer, 2004). Starting from the basic conditions presented above, Humphrey (2000) highlighted some necessary characteristics for a team to be considered an effective team: the team must be cohesive, it must have some ambitious goals, frequently receives comments on work done and should have a working framework for the tasks. In addition, globalization has led to the evolution of software development teams, and has created several types of teams with similar characteristics. Among these types are the GSD (Global Software Development) teams, geographically distributed teams, which use virtual communication channels such as email, and whose activity is focused on software development (Herbsleb & Moitra, 2001). These teams can be considered as a specification of the virtual teams (Martins et al., 2004), and are further encouraged by the relationships between customers of software development outsourcing organizations and developers (Heeks et al., 2001). GSD teams have a number of critical factors which must be considered before using a model for the management of teams in such environment. These factors are mainly the difficulty in communicating due to the geographical distances (Cramton, 2001), the cultural differences among team members (Ali Barbar et al., 2007), and the level of trust among team members (Jarvenpaa et al., 1998).

Finally, reports of TSP utilization are slim; see Callison & MacDonald (2009) for a complete bibliography. Specifically, reports of TSP utilization in distributed teams are limited to few cases; see Callison & MacDonald (2009). Distributed teams are a working style in outsourcing environments. In one of the reports of TSP utilization, Serrano & Oca (2004) assert that TSP is flexible enough to be used in outsourcing environments making some tunings but these changes were not defined. Due to the fact that these tunings are not defined across the literature, the present article aims to assess the difficulties for each activity and goal of each role defined in TSP, in order to know the tune that each role needs for outsourcing environments. The remainder of this paper is organized as follows. Firstly the issues and solutions of development teams in distributed development are presented. Secondly, TSP model is summarized. Thirdly, the difficulty of each task and goal for all roles defined in TSP is assessed using a qualitative assessment and based on the state of the art. Finally, a general overview of the assessment is discussed and conclusions are presented.

Development Teams in GSD, issues and solutions.

Software development is an intense human capital activity, more intense in intellectual capital (Sommerville & Rodden, 1996). Focusing on human capital, utilization of teams for the execution software development task means a set of advantages versus the execution these tasks individually (DeMarco & Lister, 1987). Within the working in teams advantages for software development tasks, a higher performance (Mizuno, 1983), a wide range of abilities and knowledge (Humphrey, 2000), and a higher capability for coordination (Faraj & Sproull, 2000) are the most noticeable ones. Despite of these advantages, there are problems that have to be addressed:

- Ineffective leadership; despite de fact that leadership is essential for team work (Faraj & Sambamurthy, 2006), few people are leaders by nature, so that most needs to develop leadership skills (Gorla & Jam, 2004). Among these skills, an effective team leader should motivate and inspire (Covey, 1992);
- Lack of commitment and cooperation; sometimes one or more team members may be unwilling to cooperate with the team (Grover et al., 1988);
- Differences in contributions; contributions to the overall work of the team may vary (Curtis et al., 1988);
- Lack of trust (Bandow, 2004);

In addition to the issues of software development teams, the geographical distribution inherent to virtual teams carries several problems that arise from two factors inherent to the distribution; the distance between team members and the dependence of the necessary technology to allow members to communicate (Hinds & Bailey, 2003). These two factors result in various problems: distance may result in different context (Cramton, 2002), lack of familiarity with the work habits of the rest of the team members (Goodman & Leyden, 1991), cultural diversity (Mortesa & Hinds, 2001), low confidence among the members (Jarvernpaa & Leidner, 1999), technological dependence may lead to less cohesion (Strauss & McGrath, 1994), less identification with the team (Bouas & Arrow, 1996), reduced performance work (Walther & Burgoon, 1992), exclusion of certain communications (Cramton, 2001), difficult coordination (Purdy, et al. 2000).

The problems associated with distance and reliance on technology can be minimized, once they have been identified, taking into account that they cannot be eliminated in its entirety (Olson & Olson, 2000). Among the ideas that can minimize these problems includes: increasing richness of communication channels (Daft & Lengeler, 1984), use of tools to share information in context (Atkins et al., 2002), increasing frequency and duration of face to face meetings (Kraut et al., 1992), share knowledge since the beginning of the creation of the team (Kraut et al. 1992), collaborative problem solving (Kirschner & van Bruggen, 2004).

Finally, it has to be highlighted that the use of distributed teams has its benefits, not all are issues and risks. Carmel (1999) listed six main potential benefits of distributed software development: mergers and acquisitions, position as global organizations, increase proximity to market, access to most talented developers, reduce development costs, and reduce time to market. Furthermore, managing software quality, which includes TSP, is further acerbated by the complexity of GSD as Muthuswamy & Crow pointed (2003).

Team Software Process (TSP)

Team Software Process (TSP) is a methodology that extends and refines the methods of CMM (Paulk et al., 1993) and PSP (Personal Software Process) (Humphrey, 1995) in order to guide engineers in creating and maintenance of software development teams, consisting of between 3 and 20 software engineers. TSP is based on the following principles: the software engineers are rigorous in their assignments and plan their work more precisely, making their own plans they commit the to reach team, engineers can balance the workload to meet schedules and / or minimize the duration of the project, they collect measurement data from each tasks in which are involved, the maximization of productivity go through the maximization of quality (Humphrey, 2000).

The use of TSP has the prerequisite of knowledge of PSP by all engineers, as this serves as a knowledge base about the practice of software engineering. As the author pointed in a later study, the lack of this knowledge is pointed as a critical factor for a successful implementation of the model (Humphrey, 2002). TSP main benefit is to show engineers how to produce quality products within planned costs and schedules on the go. Additionally, as indirect objective, TSP allows organizations to consolidate the CMM practices regarding the management and creation of software (Humphrey, 1998). With regard to the objectives sought by TSP, Humphrey (1998) notes the following ones. Firstly, TSP seeks to create self-managed teams who plan and track their work, set goals and have its processes and plans. Secondly, TSP shows to senior management a way to train and motivate their teams, and help them maintain peak performance. Thirdly, accelerates software process improvement by making the behavior of CMM level 5 normal and expected. In last place, TSP guides organizations toward improving high maturity.

TSP uses as a basic a project management structure based on an iterative architecture of eight phases: launch, strategy, planning, requirements, high level design, implementation, integration and testing, and post-mortem. The project can start and stop at any phase or start in the first and finish in the last phase. In all of the iterations, a launch process is carried out; this process requires a work allocation of 3 or 4 days (Humphrey 2000). During this launch process, a TSP team produces the following products (Humphrey, 1998): goals and roles for each member and for the team, a plan for the development process, a project a quality plan, a plan of support, a risk management plan and a report for project status.

TSP analysis in GSD environments

Once TSP model has been presented, firstly a scan of the state of the art for each activity of each role defined in TSPi has been carried in order to analysis the model in GSD environments. The focus during this scanning has been to look for problems or difficulties reports, and satisfactory reports for each activity. Some of these activities have been divided because its goal is multiple so the search of the references has been divided. The research of references has been based on three knowledge areas: outsourcing, as a framework; virtual teams, as a generic definition of working teams in outsourcing environments; GSD teams, as a specific definition of working teams for software development in outsourcing environments.

Once the state of the art has been analyzed for each activity, a qualitative measurement of criticality for each activity jointly with an explanation of the value has been done. The measurement uses the following scale:

- Well defined difficulties; defined difficulties for the realization of these activities have been found and widely recognized.
- Existence of difficulties; problems and difficulties for the correct realization of this activity have been found. These problems have not been
- Difficulties not found; only satisfactory references that characterize the activity as not problematic in this environment have been found.
- References not found; references that manage or mention the activity in this environment have not been founded.

All TSP roles have two activities that have to be done, that is “Participate in producing the development cycle report” and “act as a development engineer”. These activities are present separately in the section General Activities.

Team Leader

1. Motivate the team members to perform their tasks.

Motivate is one of the general tasks for a leader (Avolio & Locke, 2002). In outsourcing environments, teams are mixed, each member can be from a different organization, and therefore the ability to motivate team members becomes more important (Domberger, 1998). In addition, motivation is a risk factor in outsourcing relationships (Mehta, 2006). Taking into account these factors, this activity is assessed as ***Existence of difficulties***.

2. Run the weekly team meeting (Every Week).

Communication is one of the key factors for success in the ICT sector organizations (Perry et al., 1994) and in turn it is to maintain a relationship of Outsourcing in a satisfactory manner to all participants (Birks, 2007). Focus the problem of communication in meetings Outsourcing environments, the meetings are unable to be done face-to-face always because the movements are costly (Barthélemy, 2001), therefore team members have to use electronic media such as videoconferences, which are less richness methods for communication than face-to-face ones (Ebert & De Neve, 2001). With this information, this activity is assessed as ***Well defined difficulties***; the geographical categorization of outsourcing relationships has not be take into account (face-to-face meetings are less costly in nearshoring than in offshoring due to the closer geographical situation of members).

3. Report team status to the instructor (Every Week).

There is a main factor that influences this activity: communication. Based on the extended conclusion that communication is a key factor for a successful outsourcing relationship (Mao et al., 2008), and with the need to use electronic communication tools in this environment (Herbsleb, 2007), this activity is assessed as ***Existence of difficulties***; it can be outsourced but need some adaptations and the use of common tools for communications.

4. Help the team in allocating tasks and resolving issues.

This activity has two tasks that have been analyzed separately: allocation of tasks and resolving issues. On the one hand, the allocation of tasks in an environment of Outsourcing may resemble the one in virtual organizations in which models are used to manage and Modular Network Design (MND) that includes a step of allocating tasks (Hoogeweegen, 1999). On the other hand, the resolving issues tasks, can be explicit in the contract of Outsourcing and models for government outsourcing relationships such as eSCM (eSoucing Capability Model) provide specific practices for resolving issues based on the ideal target to build proactive and collaborative (Hyder et al., 2006, Hefley & Loesche, 2006). In contrast, it is noteworthy that the appearance of conflict environments in which team members are geographically distributed than in collocated environments, where team members are working in the same place (Jarvenpaa & Leidner, 1999). This increase in the number of conflicts is due to two factors inherent to the geographical distribution: the distance between team members and the dependence on technology to communicate between them (Hinds & Bailey, 2003).

Both tasks of this activity are successfully carried out in open source environments, which have similar characteristics with outsourcing environments: distributed and collaborative. Regarding the allocation of tasks, there is a fundamental difference between governance and Open Source environment; in Open Source environments the allocation of tasks is typically performed by a task assignment rule from bottom to top, a rule that is opposed to the organizational processes that are implanted in a governance management environments. Finally, this activity is assessed as ***Existence of difficulties***.

5. Act as facilitator and timekeeper for all team meetings.

Acting as a facilitator is vital for virtual teams (Kayworth & Leidner, 2002). Nunameker et al. (2009) argued that training teams to self-facilitate is a principle for effective virtual teamwork. Paul et al. (2004) highlighted the importance of this training. Regarding to timekeeper role (Tomei, 2007) there is no literature that can shed light, so is not taken into account for the measurement of this activity. Finally, the meetings are conducted in most cases, through electronic means such as videoconferencing, which have less communicate richness face-to-face meetings.

This activity is assessed as ***Existence of difficulties*** because of the need to train team members as self-facilitator to make virtual meetings effective.

6. Maintain the project notebook.

The ability to perform this activity is mainly based on the technology used for project notebook management. This tool needs to be available all time and at any place; for example web based tool (Alshawi & Ingirige, 2003). Therefore this activity is assessed as ***Difficulties not found*** because only a specific tool for this environment is needed in order to successfully execute this task.

Support Manager

1. Lead the team in determining its support needs and in obtaining the needed tools and facilities.

This activity can be summed as project infrastructure management which can be defined as a project infrastructure as the software, hardware, network, data, and content comprising the working environment of the project team. Nauman & Igbal (2005) pointed to a significantly reduction of risks commonly associated with the virtual project team approach if an integrated project infrastructure it’s established. In addition, if the project infrastructure provides knowledge-base capacity, it will allow team members to collaborate on and share source code, articles, lessons learned, tips & tricks, procedures, sample deliverables, and other project artifacts. Moreover, in order to successful carry this activity out, the manager should have high communicative skills and there should be good communication channels available all the time. Finally, if the development is split around the clock, this activity may be assigned to various members so team always has a project facilitator. Considering these factor, this activity is assessed as ***Existence of difficulties***.

2. Chair the configuration control board and manage the change control system.

The usage of tools for change and configuration management emerged due to the existence of parallel developments: simultaneously various people collaborate in the same development, and due to the hard difficulties for the integration of the product without these tools. These tools may be complemented with communication tools, mostly when the team members of the development are geographically dispersed and need to communicate realized changes (Bersoff, 1992).

Generally, these tools suppose the creation of an additional layer in the organizational knowledge management (Marsh & Burke, 2001). This new layer suppose an extra work effort at the project level, on the one hand, reduce the time needed for the production and integration of new versions and therefore, as long as project growth, the cost will be reduced due to a reduction of cost needed for the generation of evolutive versions of the products (Frederick, 1981). On the other hand, these tools increase the time needed for producing changes, mostly when changes are slight and versions are frequently needed.

Leading change management to outsourcing environments, two key factors have to be considered: the automatizing of change process and the delegation of the controls for change management . Karolak (1998) views responsibility and accountability to be crucial and maintains that careful consideration can avoid many of the major problems associated with GSD. Regarding this activity, Karolak's guidance for GSD teams includes a virtual software configuration control board (SCCB). In addition, the outsourcing program manager, who is sited in service provider organization, is responsible for the change control reviews and approval. In this direction, a case study of requirement evolution in an outsourcing environment used successfully a change control system as part of requirement management (Lormans et al., 2004). Moreover, McBride (2009) argued that it can be outsourced.

Finally, because of the fact that change management is not based on tools perfectly integrated on software development process, and itself is a requirement for a satisfactory implantation (Krikkar & Crnkovic, 2007), and requires the use of additional tools for communication, this activity is assessed as ***Existence of difficulties***.

3. Manage the configuration management system.

A software configuration management (SCM) tool can assist distributed projects (Lanubile, 2009) and becomes critical due to the characteristics of the distributed development (physical distance, cultural differences, trust, communication and other factors). Moreover, it can reduce the miscommunication because it enforces a common work process and a common view of the project (Carmel & Agarwal, 2001). According to Battin et al. (2001), is more challenging for global distributed projects, and a solution could be a common SCM tool with multisite replication and a centralized bug repository. Taking into account these lessons jointly with the importance of this activity in distributed environments, it is assessed as ***Existence of difficulties***.

4. Maintain the system glossary.

The maintenance of the system glossary may resemble collaborative knowledge tools such as wiki (Huettner et al., 2007). Because of this similarity, the assessment of this activity is based on the ability of such systems to manage knowledge in collaborative environments and distributed. Perhaps the best known examples of such tools are wikis, which has shown a growth of collaboration and high (Cress & Kimmerle, 2008). Because these systems can have problems if the knowledge that manage grows to large amounts of information (Eppler & Mengis, 2008), this task is assessed as ***Difficulties not found***.

5. Maintain the team's issue- and risk-tracking system.

This activity is based on two tracking systems, so the analysis is divided. On the one hand, in a study of tools for collaboration on distributed software teams, Lanubile (2009) presented some tools that can be used for issue tracking in distributed environments. In addition, Neef (2003) presented a successful case of issue-tracking system for global use. Moreover, issue-tracking systems can be accessed via www and used for learning about the projects history (Cubranic et al., 2005). Considering these findings, issue-tracking system for this environment has to be a collaborative and accessible for all members so this activity is measured with an 8. On the other hand, risks in distributed projects tends to be less visible (Karolak, 1998), and therefore more difficult to deal with. In this scenario, there are approaches to manage risk in distributed projects, to manage risk related to requirements, to manage risk in distributed IT projects (Prikladnicki et al., 2006).

Within these scenario, this task is assessed as ***Difficulties not found***, both activities can be externalized but require an extra effort to build a system that allows these managements in the specific environment.

6. Act as the team's reuse advocate.

Reuse in Outsourcing is based on the ability to encourage the reuse and address intellectual property protection problems and contracts (Kim & Stohr, 1998). Outsourcing models such as eSCM-CL (eSourcing Capability Model for Client Organizations) includes a practice for the management of each of these tasks: Intellectual Property thr03 and thr04 Security & Privacy (Hefley & Loesche, 2006). Similarly the model eSCM-SP (eSourcing Capability Model for Service Providers) include: Intellectual Property thr05 and thr06 Statutory & Regulatory Compliance (Hyder et al., 2006). Based on the existence of good practices to address the problems related to reuse in this environments, jointly with the need to encourage this activity, and the independence reuse degree with respect to organizational model (Lynex & Layzell, 1998), a value of ***Existence of difficulties*** is obtained for the assessment of this practice.

Planning Manager

1. Lead the team in producing the task plan for the next development cycle.

Planning in virtual teams has more difficulties in planning activities than collocated teams. It can be explained by these additional four elements: coordination requirements, resource constraints, responsibility for progress, mapping of tasks among the members (Cascio & Shurygailo, 2003). On the one hand, the improvement in communication tools has led organizations to assign tasks to distributed groups of workers as opposed to collocated groups (Jang et al., 2000). On the other hand, the importance of a coherent distribution of tasks is a critical activity for better performance in GSD (Ebert & De Neve, 2001). Due to its criticality and because this activity can be outsourced this activity is assessed as ***Existence of difficulties***.

2. Lead the team in producing the schedule for the next development cycle.

The difficulty in achieving a schedule in environments with workers split in multiple location is higher than single site environments (Cascio & Shurygailo, 2003) due to the necessity of relying on electronic means to communicate (Jang et al., 2000) and the existence of temporal differences that reduces the available time for communications (Solomon, 1995). For these reasons, this activity is assessed as ***Existence of difficulties***.

3. Lead the team in producing the balanced team plan.

One of the main motivators for the use of distributed teams is the ability to distribute the workload among different sites (Anschuetz, 1998), that is why the activity is assessed with ***Difficulties not found***. This punctuation is given due to the existence of factors such as trust which hinder the measured activity (Moe & Smit, 2008).

4. Track the team's progress against the plan.

As Bell and Kozlowski (2002) argued the main issue for tracking and monitoring in virtual teams is the necessity of distributing the function to the team itself. In addition, the lack of face-to-face meetings restricts the ability to track progress. On the other hand, the self-directed characteristic of TSP teams reduces the criticality of this decision. Due to these factors, this activity is assessed as ***Existence of difficulties*** because it can be externalized, but some issues that have to be addressed.

Development Manager

1. Lead the team in producing the development strategy.

According to Casey & Richardson (2006), to be effective a successful project management virtual software team, the strategy must address the specific needs of this dynamic environment. In addition, the authors argued that "the objective may be to implement a sustained virtual software team strategy to leverage the technical experience of staff at one location with the availability of competitive cost engineers in another". In a case study of geographically distributed teams carried out by Evaristo et al. (2004) a Japan company used a development strategy that involves a "hands-on approach, with small teams being deployed into the client's facilities and keeping in touch with headquarters only for a limited amount of know-how and general directions". McCaffery et al. (2006) reported a case study of a change in software development strategy in order to continue as a significant player within the GSD industry. Considering that outsourcing represents outsourcing represents a development strategy by itself, which is not a straightforward task, and considering the cases studies found, this activity is assessed as ***Well defined difficulties***; it need to be keep in client side unless full development it outsourced.

2. Lead the team in producing the preliminary size and time estimates for the products to be produced.

According to Parthasarathy (2007) the environment in which outsourcing service provides organizations do estimations is competitive and complicates the estimation process, but don't point out any limitation or difficulty for carrying this activity out. In addition, Zanoni and Audy (2004) identified size estimation as more difficult task in distributed projects due to requirement specification, test process and communication between participants being more difficult. Therefore, this activity is assessed as ***Existence of difficulties***.

3. Lead the development of the requirements specification (SRS).

The analysis of software requirements in global virtual teams carried out by Edwards & Sridhar (2005) indicates that this activity can be done in outsourcing environments. On the other hand, Lormas et al. (2004) reported some issues and problems in the management of software requirements evolution in outsourcing context. Moreover, Smite (2006) obtained some lessons from requirement management in GSD teams that can be useful for this environment, and indicated that this task is arduous if the process is not well defined and if the team is not prepared to work in this scenario. These lessons include for example training in soft skills, use of communication tools, and mutual knowledge. Taking into account these factors, this activity is assessed as ***Existence of difficulties*** because it can be outsourced but the mentioned issues need to be addressed.

4. Lead the team in producing the high-level design.

The production of high-level designs can be considered as strategic function; therefore the difficulty for outsourcing it is high and can be easier to do it on mature outsourcing relationships (Fisher et al., 2008). In this direction, there are recommendations for backing in-house the high-level designs (Tiwana et al., 2008) and are pointed as a negotiation item with vendors (Lormans et al., 2004). Considering the found recommendations, this activity is assessed as ***Well defined difficulties***.

5. Lead the team in producing the system design specification (SDS).

The system design specification can be created with content management software in GSD environments, but are limited when coediting a complex document (Al-Asmari & Yu, 2006). Lormans et al. (2004) carried a study of a outsource relationship which successfully included design steps from SRS to SDS, therefore this activity is assessed as ***Existence of difficulties***.

6. Lead the team in implementing the product.

According to Zigurs (2003) virtual teams provide a unique opportunity for redefining the concept of leadership. Casey & Richardson (2006) concluded that project manager must address the specific needs of the dynamic environment. In addition, it is quite common for organizations to meet their software from outsourcing vendors (Wybo, 2007). The factors that influence this activity are the general ones: communication-distance, cultural differences, trust, etc. (House et al., 2008). With this information, this activity is assessed as ***Difficulties not found***.

7. Lead the team in developing the build, integration, and system test plans.

This activity is three-fold so the analysis has been split. Firstly, according to McMahon (2001) planning and coordinating the builds across distributed sites may be the single greatest challenge faced on virtual projects; therefore this activity is measured with a -5. Secondly, the integration plans need to minimize the dependencies across sites (Herbsleb & Grinter, 1999). In a case-study of reference models for software product integration that included distributed development (Larsson et al., 2009) pointed to four class of problems: inadequate selection and implementation of strategy, inadequate management of architecture and design, inadequate establishment or use of the integration environment, inadequate delivery of functions. For example a share integration plan in order to share points for integration and tasks may be a solution for distributed teams (Cameron, 2006). Another approach is to use an incremental integration plan (Battin et al., 2001). Kommeren & Parviainen (2007) indicate that lack of an explicit integration plan can cause inefficiency and extra complexity to integration activities.

Lastly, the system test plans may be designed for allowing dependency of task distribution (Taweel & Brereton, 2006) and following a strategy that provides predictions before the code is implemented and be capable of taking into account new information about the risk at any stage in development. Therefore, this activity is assessed as ***Existence of difficulties***.

8. Lead the team in developing the test materials and running the tests.

In a case study, Leszak and Meier (2007) presented a successful case of a large-scale global systems & software development that included software testing. In another case, Hornett (2004) reported a successful case of a virtual team that has the purpose to create and test software to ensure transferability of documents. There are no case of failures neither of problems with this activity in such environment, therefore this activity is assessed as ***Difficulties not found***.

9. Lead the team in producing the product's user documentation.

According to a student study carried out by Ellis (2006), interpersonal and intrapersonal interaction and communication among or between team members is needed in order to successful produce documentation. These factors can be moved to product’s user documentation and be a critical factor in outsourcing environments due to the inherit problems of communication and coordination. Another key factor for producing documentation in distributed environments is the use of a shared repository (Jang et al., 2000). Taking into account these findings, this activity is assessed as ***Existence of difficulties***; it can be outsourced but need a huge extra effort.

Quality-Process Manager

1. Lead the team in producing and tracking the quality plan.

In a case study about pitfalls in remote team coordination, Smite et al. (2008) indicated that using a quality system was implemented and process were certified to ISO 9001:2000 standard, and jointly with weekly teleconferences to discuss urgent problems and plans, the project followed standardized guidelines for requirement specification, task management, progress reporting and monitoring, and other activities. In addition, Prikladnicki and Audy (2003) argued that the implementation of a quality model and a continuous software process improvement are very important to succeed. In other case about a distributed project, Spinellis (2006) pointed that there are no apparent ill effects on the quality of code. This lack of difference between collocated and distributed is not reflected in the study carried out by Canfora et al. (2006). In this study, the results pointed to a link between quality and some factors of distribution settings. Such factors are coordination, communication, and development site (Raffo & Setamanit, 2005). Moreover, organizations search for competitive advantages in terms of cost, quality and flexibility in the area of software development (Prikladnicki et al. 2002), and are a motivator for investing in global software development (Prikladnicki et al., 2003). In a case study of distributed software development in offshoring and onshoring (Prikladnicki et al., 2007), the authors highlighted that the main challenges for quality management were related to difficulties in configuration management, not having a common software process among the distributed teams, and that most of the knowledge was concentrated in the supplier company, making the project documentation hard for the client organization. Considering these findings, this activity is assessed as ***Existence of difficulties***.

2. Alert the team, the team leader, and the instructor to quality problems.

Quality problems in GSD projects are not uncommon (Carmel, 1999), therefore the alert and communication of quality problems is needed in distributed environments too. Moreover, the management of quality has lead to the creation of excellence models such as eSCM (Hyder et al., 2006, Hefley & Loesche, 2006) which include quality management practices. In GSD, the channels used for communicate and alert are virtual ones instead of face-to-face, and therefore they have specific issues that need to be addressed in order be effective (Anderson et al., 2007). Considering these findings, this activity is assessed as ***Existence of difficulties***.

3. Lead the team in defining and documenting its processes and in maintaining the process improvement process.

This activity may be threatened by differences in organizational cultures and management practices so the internal procedures may vary. These differences between software processes used by collaborating groups and associated process maturities may be considered as a risk category for virtual development projects (Maidantchik & da Rocha, 2002). In addition to these findings, McMahon (2001) warned about the potential danger of differences in processes and process maturities of collaborating parties in distributed development. Therefore, this activity is have risks and is assessed as ***Existence of Difficulties***.

4. Establish and maintain the team's development standards.

One of the main reasons of using development standards is quality assurance (Pressman, 2001). Establishing a standard for development in GSD may be done by balancing plan-driven and agile methodologies (Moe & Smite, 2008) for project management level, and for the level of development standards may be established and used but there have not been found references about this issue. This activity is assessed as ***Existence of difficulties***.

5. Review and approve all products before submission to the CCB.

Change management in software developments represents a critical key for unique site developments. In GSD environments, it is necessary to redefine the process in order to adjust to the specific peculiarities of these environments and the synchronization between projects can support change management and make it transparent (Carmel, 1999). The centralization of revision and acceptation tasks about products before been communicated to CCB reduces some of the lessons learned about software change management in distributed environments (Pilatti et al., 2006): working in a unique environment of configuration management and defining a coordinator for products elaboration.

Considering that this activity is not realized automatically in centralized environments and present some problems in distributed environments, this activity is assessed as ***Existence of Difficulties***.

6. Act as team's inspection moderator.

The goal of this activity is the identification of the defects of each team member's products. During the meeting, the moderator has to focus the attention on the identification avoiding the discussions about the fix actions for each defect. So considering the meeting problems described in the GSD literature (non face-to-face, cost in travels, etc…), the specific problem for this activity is the difficulty for moderator to focus on the identification instead of the fixing of the defects. After a deep analysis of research literature there have been not found references that treat this activity in GSD environments, therefore this activity is assessed as ***References not found***.

7. Act as recorder in all the team's meetings.

In GSD environments, meetings are mainly virtual ones and communication channels are based on technologies rather than face-to-face meetings (Anderson et al., 2007). Considering the inherent characteristics of communication technologies used for these meetings, the activity of recording the team's meetings may not represent any difficulties, and therefore this activity is assessed as ***Difficulties not found***.

General Activities

1. Participate in producing the development cycle report.

The goal of development cycle report is to establish the grade of conclusion of planned work for each team participant and to identify possible improvements in future plans and similar works. Each role participate identifying which tasks have been done, which of the allocated and planned tasks have not been done for the present development cycle, and the possible improvements of these tasks. The participation on this activity is proactive because of the improvement focus of the activity for the next cycles and future projects. Based on these characteristics, the assessment of this activity is based on the proactive style jointly with the ability to generate and obtain information about the project state with participative actions.

On the one hand, proactive style in outsourcing relations is reached once the contract and the SLA down to a second dimensions and a win-win relationship is the first dimension (Lee et al., 2003). In this direction, models for government outsourcing relationships such as eSCM (eSoucing Capability Model) promote and ideal of proactive and collaborative style (Hyder et al., 2006, Hefley & Loesche ,2006). On the other hand, generating and obtaining information represent the background of GSD and there are assumed benefits such as cost reduction, cross-site modularization of development work, and the access to large skilled labor pool, closer proximity to market and customer (Conchúir et al., 2009) that makes GSD attractive for organizations if associated risks are mitigated.

Considering the presented issues, this activity is assessed as ***Well defined difficulties***.

2. Act as a development engineer.

This activity may be considered as the global activity for GSD teams; therefore the problems and issues joint with their solutions are the inherent ones for GSD, which have been quickly glanced across the present article, and are widely treated in research literature. Taking into account this conclusion, this activity is assessed as ***Well defined difficulties***.

Discussion

Once all the activities of each TSP role have been assessed and in order to sum up the obtained results, a table that summarizes each TSP role assessment are presented and discussed. Firstly, team leader's assessment results point to a general existence of difficulties which may be interpreted as an extra effort in order to carry this role in GSD environments, or an incompatibility between this role and GSD.

Team Leader	
Activity	**Assessment Result**
Motivate the team members to perform their tasks	Existence of difficulties
Run the weekly team meeting (Every Week)	Well defined difficulties
Report team status to the instructor (Every Week)	Existence of difficulties
Help the team in allocating tasks and resolving issues	Existence of difficulties
Act as facilitator and timekeeper for all team meetings	Existence of difficulties
Maintain the project notebook	Difficulties not found

Table 1. Team Leader Assessment

Secondly, support manager's assessment results point to a less existence of difficulties and there are no well defined difficulties which may be interpreted as a possibility to d carry this role in GSD environments with a little extra effort in managing these activities.

Support Manager	
Activity	**Assessment Result**
Lead the team in determining its support needs and in obtaining the needed tools and facilities	Existence of difficulties
Chair the configuration control board and manage the change control system	Existence of difficulties
Manage the configuration management system	Existence of difficulties
Maintain the system glossary	Difficulties not found
Maintain the team's issue- and risk-tracking system	Difficulties not found
Act as the team's reuse advocate	Existence of difficulties

Table 2. Support Manager Assessment

Thirdly, planning manager's assessment results seem parallel to support manager's which point to a little extra effort in managing theses activities.

Planning Manager	
Activity	**Assessment Result**
Lead the team in producing the task plan for the next development cycle	Existence of difficulties
Lead the team in producing the schedule for the next development cycle	Existence of difficulties
Lead the team in producing the balanced team plan	Difficulties not found
Track the team's progress against the plan	Existence of difficulties

Table 3. Planning Manager Assessment

Fourthly, development manager's assessment results have at least one result in each scale value. Assuming that having difficulties superimpose the non existence of its, the results points to an extra effort in order to carry this role in GSD environments.

Development Manager	
Activity	**Assessment Result**
Lead the team in producing the development strategy	Well defined difficulties
Lead the team in producing the preliminary size and time estimates for the products to be produced	Existence of difficulties
Lead the development of the requirements specification (SRS)	Existence of difficulties
Lead the team in producing the high-level design	Well defined difficulties
Lead the team in producing the system design specification (SDS)	Existence of difficulties
Lead the team in implementing the product	Difficulties not found
Lead the team in developing the build, integration, and system test plans	Existence of difficulties
Lead the team in developing the test materials and running the tests	Difficulties not found
Lead the team in producing the product's user documentation	Existence of difficulties

Table 4. Planning Manager Assessment

Finally, quality-process manager's results follow the results obtained for all roles which points to a general presence of difficulties so there is an extra effort needed in order to carry this role in GSD environments. There have not been found references for acting as team inspector moderator so it has been not assessed, but this lack of assessment doesn't influence the general result of this role.

Quality-Process Manager	
Activity	**Assessment Result**
Lead the team in producing and tracking the quality plan	Existence of difficulties
Alert the team, the team leader, and the instructor to quality problems	Existence of difficulties
Lead the team in defining and documenting its processes and in maintaining the process improvement process	Existence of difficulties
Establish and maintain the team's development standards	Existence of difficulties
Review and approve all products before submission to the CCB	Existence of difficulties
Act as team's inspection moderator	References not found
Act as recorder in all the team's meetings	Difficulties not found

Table 5. Quality-Process Manager Assessment

To sum up the results some figures are presented: a figure that represents the assessment for each role is presented (Figure 1), and a figure that represents the sum up of the assessment results (Figure 2).

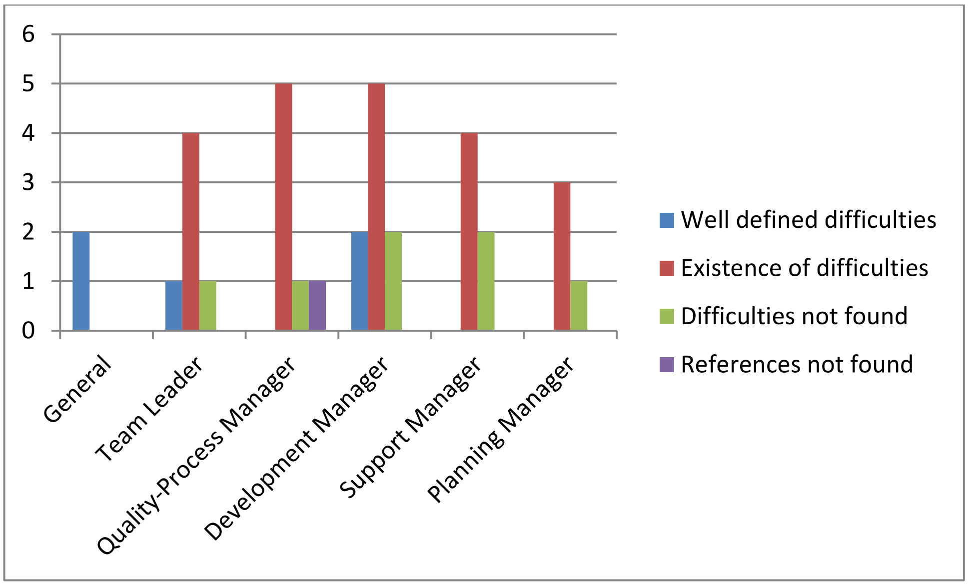

Figure 1. Assessment results for each role

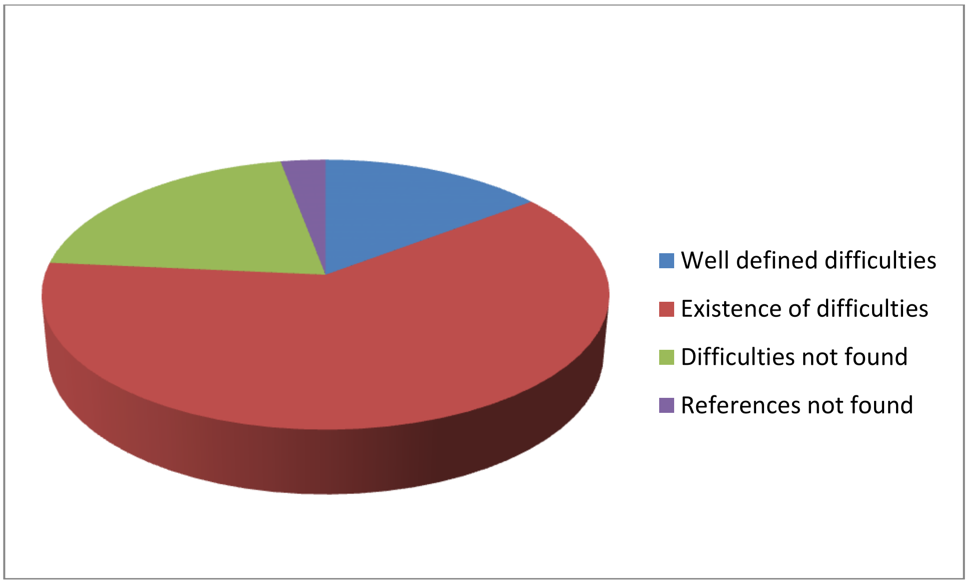

Figure 2. Assessment results

Conclusions

In this the paper, the authors have found several difficulties in order to find references that treat the activities in outsourcing, GSD, or virtual team environments. These difficulties have lead to a superficial assessment which doesn't have to be considered as the truth, but they can shed light for the utilization of TSP in these environments. On the one hand, considering the results for each role, there is not a role which have no difficulties and all roles have at least a activity that have been assessed as difficulties not found. These results point to a domination of the difficulties. On the other hand, considering the obtained results as global results, more than 75% of the activities of TSP have difficulties in order to execute them in GSD environments. This result may indicate a difficulty in using TSP in these environments which have to be addressed in future researches with the purpose of assessing the calibration needed for each difficulty. In addition to the strong presence of difficulties for the activities, there is a little presence of difficulties not found assessments point to the same direction.

Finally, the authors propose two future works: (1) calibration of each activity in these environments, which may be done individually for each task or grouping them into categories and may lay the foundation stone for the creation of a Virtual TSP; (2) contrast the obtained results in distributed projects with the purpose of validating the results.

References

Acuña, S. T., & Juristo, N. (2004). Assigning people to roles in software projects. *Software - Practice & Experience*, 34(7), 675-696.

Al-Asmari, K. R., & Yu, L. (2006). Experiences in Distributed Software Development with Wiki. In Hamid R. Arabnia, Hassan Reza (Eds.), *Proceedings of the International Conference on Software Engineering Research and Practice & Conference on Programming Languages and Compilers* (pp. 389-293). Las Vegas, Nevada: CSREA Press.

Ali Babar, M., Verner, J. M., & Nguyen, P. T. (2007). Establishing and maintaining trust in software outsourcing relationships: An empirical investigation. *Journal of Systems and Software*, 80(9), 1438-1449.

Alshawi, M., & Ingirige, B. (2003). Web-enabled project management: an emerging paradigm in construction. *Automation in Construction*, 12(4), 349-364.

Anderson, A. H., McEwan, R., Bal, J., & Carletta, J. (2007). Virtual team meetings: An analysis of communication and context. *Computers in Human Behavior*, 23(5), 2558-2580.

Anschuetz, L. (1998). Managing Geographically Distributed Teams. *IEEE Professional Communication Conference (IPCC)* (pp. 23-25). Quebec City, Canada: IEEE International.

Atkins, D., Boyer, D., Handel, M., Herbsleb, J., Mockus, A. & Wills, G. (2002). *Achieving speed in globally distributed project work*. Paper presented at the Human Computer Interaction Consortium, Winterpark, CO.

Avolio, B. J., & Locke, E. E. (2002). Contrasting different philosophies of leader motivation: Altruism versus egoism, *The Leadership Quarterly*, 13(2), 169-191.

Bandow, D. (2004). Time to create sound teamwork. *Journal for Quality and Participation*, 24(2), 41-47.

Barthélemy, J. (2001). The hidden cost of IT outsourcing. *MIT Sloan Management*, 42(3), 60-69.

Battin, R. D., Crocker, R., Kreidler, J., & Subramanian, K. (2001). Leveraging resources in global software development. IEEE Software, 18(2), 70-77.

Bell, B. S., & Kozlowski, S. W. J. (2002). A topology of Virtual Teams: Implications for Effective Leadership. *Group & Organization Management*, 24(14), 14-49.

Bersoff, V., & Chase, R. (1982). *Software Configuration Management - An investment in Product Integrity*. Englewood Cliffs, N.J.: Prentice Hall Inc.

Birks, D. F., Zainuddin, Y., Choo, A. T., Wafa, S. A., Morar, S., & Nasirin, S. (2007). Successful IT Outsourcing Engagement: Lessons from Malaysia. *The Electronic Journal of Information Systems in Developing Countries*, 30(5), 1-12.

Blackburn, J. D., Scudder, G. D., & Van Wassenhove, L. N. (1996). Improving speed and productivity of software development: a global survey of software developers. *IEEE Transactions on Software Engineering*, 22(12), 875-885.

Huettner, B., Brown, M. K., & James-Tanny, C. (2007). *Managing virtual teams: Getting the most from wikis, blogs, and other collaborative tools*. Plano, TX: Wordware Publishing, Inc.

Bouas, K., & Arrow, H. (1996). The development of group identity in computer and face-to-face groups with membership change. *Computer Supported Cooperative Work*, 4(2-3), 153-178.

Callison, R., & MacDonald, M. (2009). *A Bibliography of the Personal Software Process (PSP) and the Team Software Process (TSP)* (Tech. Rep. CMU/SEI-2009-SR-025). Pittsburgh, PA:

Carnegie Mellow University, Software Engineering Institute.

Cameron, J. (2006). Governance structure, mechanisms and methods for managing collaborative eBusiness projects. *Journal on Chain and Network Science*, 6(2), 155-174.

Canfora, G., Cimitile, A., Di Lucca, G. A., & Visaggio, C. A. (2006). How distribution affects the success of pair programming. *International Journal of Software Engineering and Knowledge Engineering*, 16(2), 293-313.

Carmel, E. (1999). *Global Software Teams: collaborating across borders and time zones*. Upper Saddle River, NJ: Prentice Hall.

Carmel, E., & Agarwal, R. (2001). Tactical Approaches for Alleviating Distance in Global Software Development. *IEEE Software*, 18(2), 22-29.

Cascio, W. F., & Shurygailo, S. (2003). E-Leadership and Virtual Teams. *Organizational Dynamics*, 31(4), 362-376.

Casey, V., & Richardson, I. (2006a). Project Management within Virtual Software Teams. *Proceedings of the IEEE international conference on Global Software Engineering* (pp. 33-42) Washington, DC: IEEE Computer Society.

Colomo-Palacios, R., Tovar-Caro, E. Garcia-Crespo, A., & Gomez-Berbis, M. J., (2010). Identifying Technical Competences of IT Professionals. The Case of Software Engineers. *International Journal of Human Capital and Information Technology Professionals. 1*(1), 31-43.

Conchúir, E. Ó., Ågerfalk, P. J., Olsson, H. H., & Fitzgerald, B. (2009). Global software development: where are the benefits?. *Communications of the ACM*, 52(8), 127-131.

Cramton, C. D. (2001). The mutual knowledge problem and its consequences for dispersed collaboration. *Organizational Science*, 12(3), 346-371.

Cramton, C. D. (2002). Attribution in distributed work groups. In P. J. Hinds, S. Kiesler, (Eds.), *Distributed Work* (pp. 91-212). Cambridge, MA: MIT Press.

Cress, U., & Kimmerle, J. (2008). A systemic and cognitive view on collaborative knowledge building with wikis. *International Journal of Computer-Supported Collaborative Learning*, 3(2), 105-122.

Cubranic, D., Murphy, G. C., Singer, J., & Booth, K. S. (2005). Hipikat: a project memory for software development. *IEEE Transactions on Software Engineering*, 31(6), 446-465.

Curtis, B., Krasner, M. I., & Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM*, 31(11), 1268-1287.

DeMarco, T., & Lister, T. (1987). *Peopleware: productive projects and teams*. New York, NJ: Dorset House Publishing Co.

Domberger, S. (1999). *The contracting organization: A strategic guide to outsourcing*, New York, NJ: Oxford University Press.

Ebert, C., & DeNeve, P. (2001). Surviving global software development. *IEEE Software*, 18(2), 62-69.

Edwards, K., & Sridhar, V. (2005). Analysis of Software Requirements Engineering Exercises in a Global Virtual Team Setup. *Journal of Global Information Management*, 13(2), 21-41.

Ellis, H. J. C. (2006). An evaluation of learning in an online project-based web application design and development course. *Journal of Computing Sciences in Colleges*, 21(6), 217-227.

Eppler, M. J., & Mengis, J. (2004). The Concept of Information Overload - A Review of Literature from Organization Science, Accounting, Marketing, MIS, and Related Disciplines. *The Information Society: An International Journal*, 20(5), 1-20.

Evaristo, J. R., Scudder, R., Desouza, K. C., & Sato, O. (2004). A dimensional analysis of geographically distributed project teams: a case study. *Journal of Engineering and Technology Management*, 21(3), 175-189.

Faraj, S., & Sambamurthy, V. (2006). Leadership of information systems development projects. *IEEE Transactions on Engineering Management*, 53(2), 238-249.

Faraj, S., & Sproull, L. (2000). Coordinating expertise in software development teams. *Management Science*, 46(12), 1554-1568.

Fisher, J., Hirschheim, R., & Jacobs, R. (2008). Understanding the outsourcing learning curve: A longitudinal analysis of a large Australian company. *Information Systems Frontiers*, 10(2), 165-178.

Gannon, J. D. (1979). Human factors in Software Engineering. *Computer*, 12(12), 6-7.

Goodman, P. S., & Leyden, D. P. (1991). Familiarity and group productivity. *Journal of Applied Psychology*, 76(44), 578-586.

Gorla, N. & Lam, Y. W. (2004). Who Should Work with Whom?, Building Effective Software Project Teams. *Communications of the ACM*, 47(6), 79-82.

Grover, V., Lederer, A. L., & Sabherwal, R. (1988). Recognizing the Politics of MIS. *Information and Management*, 14(3), 145-156.

Heeks, R., Krishna, S., Nicholsen, B., & Sahay S. (2001). Synching or sinking: Global software outsourcing relationships. *IEEE Software*, 18(2), 54-60.

Hefley, W. E., & Loesche E. A. (2006). *eSourcing Capability Model for Client Organizations (eSCM-CL) v.1.1*. Pittsburgh, Pennsylvania: Carnegie Mellon University, IT Services Qualification Center.

Herbsleb, J. D., & Grinter, R. E. (1999). Architectures, coordination, and distance: Conway's law and beyond. *IEEE Software*, 16(5), 63-70.

Herbsleb, J. D., & Moitra, D. (2001). Global software development. *IEEE Software*, 18(2), 16-20.

Herbsleb, J. D. (2007). Global Software Engineering: The Future of Socio-technical Coordination. *International Conference on Software Engineering, 2007 Future of Software Engineering* (pp. 188-198). Washington, DC: IEEE Computer Society.

Hinds, P. J., & Bailey, D. E. (2003). Out of sight, out of synch: Understanding conflict in distributed teams. *Organization Science*, 14(6), 615-632.

Hoogeweegen, M., Teunissen, W., Vervest, P., & Wagenaar, R. (1999). Modular network design: using information and communication technology to allocate production tasks in a virtual organization. *Decision Sciences*, 30(4), 1073-1103.

House, D., de Vreede, G.J., Wolcott, P., & Dick, K. (2008). Success Factors for the Global Implementation of ERP/HRMS Software. In C. Ferran, R. Salim (Eds.), *Enterprise Resource Planning for Global Economies: Managerial Issues and Challenges* (pp. 289-307). Hershey, PA: Information Science Reference.

Humphrey, W.S. (1995). Introducing the personal software process. *Annals of Software Engineering*, 1(1), 311-325.

Humphrey, W.S. (1998). Three Dimensions of Process Improvement. Part III: The Team Software Process. *Crosstalk*, 11(4), 14-17.

Humphrey, W.S. (2000). *Introduction to the Team Software Process*. Reading, MA: Adison-Wesley.

Humphrey, W. S. (2002). Three Process Perspectives: Organizations, Teams, and People. *Annals of Software Engineering*, 14(1-4), 39-72.

Hyder, E. B., Heston, K. M., & Paulk, M. C. (2006). *The sCM-SP v2.01: The eSourcing Capability Model for Service Providers (eSCM-SP) v2.01, Part 2: Practice Details*. Pittsburgh, Pennsylvania: Carnegie Mellon University, IT Services Qualification Center.

Jang, C. Y., Steinfield, C., & Pfaff, B. (2000). Supporting awareness among virtual teams in a web-based collaborative system: the teamSCOPE system. *ACM SIGGROUP Bulletin*, 21(3), 28-34.

Jarvenpaa, S. L., & Leidner, D. E. (1999). Communication and trust in global virtual teams. *Organization Science*, 10(6), 791-815.

Jarvenpaa, S. L., Knoll, K., & Leidner, D. E. (1998). Is Anybody Out There?: The Antecedents of Trust in Global Virtual Teams. *Journal of Management Information Systems*, 14(4), 29-64.

Karolak, D. W. (1998). *Global Software Development - Managing Virtual Teams and Environments*. Los Alamitos, CA: IEEE Computer Society Press.

Kayworth, T. R., & Leidner, D. E. (2002). Leadership Effectiveness in Global Virtual Teams, *Journal of Management Information Systems*, 18(3), 7-40.

Kim, Y., & Stohr, E.A. (1998). Software Reuse: Survey and Research Directions. *Journal of Management Information Systems*, 14(4), 113-147.

Kirschner, P. A., & van Bruggen, J. (2004). Learning and understanding in virtual teams. *CyberPsychology & Behavior*, 7(2), 135-139.

Kommeren, R., & Parviainen, P. (2007). Philips experiences in global distributed software development. *Empirical Software Engineering*, 12(6), 647-660.

Krikkar, R., & Crnkovic, I. (2007). Software Configuration Management. *Science of Computer Programming*, 65(3), 215-221.

Krishnan, M. S. (1998). The role of team factors in software cost and quality: an empirical analysis. *Information Technology and People*, 11(1), 20-35.

Kraut, R. E., Fish, R. S., & Chalfonte, B. (1992). Task requirements and media choice in collaborative writing. *Human Computer Interaction*, 7(4), 375-407.

Lanubile, F. (2009). Collaboration in Distributed Software Development. *Lecture Notes in Computer Science*, 5413, (174-193).

Laughery Jr., K. R., & Laughery Sr., K. R. (1985). Human factors in software engineering: a review of the literature. Journal of Systems and Software, 5(1), 3-14.

Lee, J.-N., Huynh, M. Q., Kwok, R. C.-W., & Pi, S.-M. (2003). IT Outsourcing Evolution - Past, Present, and Future. *Communications of the ACM*, 44(5), 84-85.

Leszak, M., & Meier, M. (2007). Successful Global Development of a Large-scale Embedded Telecommunications Product. In Second IEEE International Conference on Global Software Engineering (pp. 23-32). Washington, DC: IEEE Computer Society.

Lerner, J., & Tirole, J. (2002). Some simple economics of open source. The Journal of Industrial Economics, 50(2), 197-234.

Lormas, M., van Dijk, H., van Deursen, A., Nocker, E., & de Zeeuw, A (2004). Managing evolving requirements in an outsourcing context: an industrial experience report. In *Proceedings 7th International Workshop on Principles of software Evolution* (149-158). Washington, DC: IEEE Computer Society.

Lynex, A., & Layzell, P. J. (1998). Organisational considerations for software reuse. *Annals of Software Engineering*, 5(1), 105-124.

Mahmood, A. M., Pettingell, K. J., & Shaskevich, A. I. (1996). Measuring productivity of software projects: A data envelopment analysis approach. *Decision Science*, 27(1), 57-80.

Maidantchik, C., & da Rocha, A. R. C. (2002). Managing a worldwide software process. *Proceedings International Workshop on Global Software Development (ICSE 2002), Orlando, FL:*.

Mao, J. Y., Lee, J. N., & Deng, C.P. (2008). Vendors' perspectives on trust and control in offshore information systems outsourcing. *Information & Management*, 45(7), 482-492.

Martins, L. L., Gilson, L. L., & Maynard, M. T. (2004). Virtual Teams: What Do We Know and Where Do We Go From Here? *Journal of Management*, 30(6), 805-835.

Marsh, G., & Burke, M. (2001). Knowledge Management and Organizational Effectiveness: balancing the mild, the wild and the crazy. *Australasian Journal of Information Systems*, 9(1), pp 67-79.

McBride, N. (2009). Exploring service issues within the IT organization: Four mini-case studies. *International Journal of Information Management*, 29(3), 237-243.

McMahon, P. E. (2001). Distributed Development: Insights, Challenges, and Solutions. *CrossTalk*, 15(11), 4-9.

McCaffery, F., Smite, D., Wilkie, F. G., & McFall, D. (2006). A proposed way for european software industries to achieve growth within the global marketplace. *Software Process Improvement and Practice*, 11(3), 277-285.

Metha, A., Armenakis, A, Mehta, N., & Irani, F. (2006). Challenges and opportunities of business process outsourcing in India. *Journal of Labor Research*, 27(3), 324-338.

Mizuno, Y. (1983). Software Quality Improvement. *Computer*, 16(3), 66-72.

Moe, N. B., & Smite, D. (2008). Understanding a lack of trust in global software development: a multiple case study. *Software Process Improvement and Practice*, 13(3), 217-231.

Mortensen, M., & Hinds, P.J. (2001). Conflict and shared identity in geographically distributed teams. *International Journal of Conflict Management*, 12(3), 212-238.

Muthuswamy, B., & Crow, G. B. (2003). Global Software Development: Strategic Implications for U.S. Information Systems Academic Programs. *International Association for Computer* Information Systems, 4(1), 271-276.

Nauman, S., Igbal, S. (2005). Challenges of virtual project management in developing countries. *IEEE International Engineering Management Conference* (pp. 579-583). New Foundland, Canada: IEEE International.

Neef, D. (2005). Managing corporate risk through better knowledge management. The Learning Organization, 12(2), 112-124.

Nunamaker, J. F., Reinig, B. A., & Briggs, R. O. (2009). Principles for effective virtual teamwork. *Communincations of the ACM*, 52(4), 113-117.

Olson, G. M., & Olson, J. S. (2000). Distance matters. *Human Computer Interaction*, 15(2), 139-178.

Parthasarathy, M. A. (2007). *Practical Software Estimation: Function Point Methods for Insourced and Outsourced Projects*. Boston, MA: Addison-Wesley Professional.

Paul, S., Seetharaman, P., Samarah, I., & Mykytyn, P. P. (2004). Impact of heterogeneity and collaborative conflict management style on the performance of synchronous global virtual teams. *Information & Management*, 41(3), 303-321.

Paulk, M. C., Curtis, B., Chrissis, M. B., & Weber, C. V. (1993). *Capability Maturity Model for Software Version 1.1* (Tech. Rep. CMU/SEI-93-TR-24). Pittsburgh, Pennsylvania: Carnegie Mellon University, Software Engineering Institute.

Perry, D. E., Staudenmayer, N. A., & Votta, L. G. (1994). People, Organizations, and Process

Improvement. *IEEE Software*, 11(4), 36-45.

Pilatti, L., Audy, J. L. N., & Prikladnicki, R. (2006). Software configuration management over a global software development environment: lessons learned from a case study. *Proceedings of the 2006 international workshop on Global software development for the practitioner* (pp. 45-50). New York, NY: ACM.

Pressman, R. S. (2001). Software engineering: a practioner's approach. New York, NJ: McGraw-Hill.

Prikladnicki, R., & Audy, J. N. L. (2003). Requirements Engineering in Global Software Development: Preliminary Findings from a Case Study in a SW-CMM context. *Proceedings of the 5th Simpósio Internacional de Melhoria de Processo de Software*. Pernambuco.

Prikladnicki, R., Audy, J. N. L., Damian, D., & de Oliveria, T. C. (2007). Distributed Software Development: Practices and challenges in different business strategies of offshoring and onshoring, *Second IEEE International Conference on Global Software Engineering* (pp. 262-274).

Prikladnicki, R., Audy, J. N. L., & Evaristo, R. (2003). Global Software Development in Practice Lessons Learned. *Software Process Improvement and Practice*, 8(4), 267-281.

Prikladnicki, R., Evaristo, R, Audy, J. L. N., & Yamaguti, M. H. (2006). Risk Management in Distributed IT Projects: Integrating Strategic, Tactical, and Operational Levels. *International Journal of e-Collaboration*, 2(4), 1-18.

Prikladnicki R, Peres, F., Audy, J. N. L., M´ora, M. C., & Perdigoto, A. (2002). Requirements specification model in a software development process inside a physically distributed environment. *Proceedings of ICEIS,* (pp. 830-834), Ciudad Real, Spain.

Raffo, D., & Setamanit, S. (2005). A Simulation Model for Global Software Development Project. *Proceedings of the 2006 international workshop on Global software development for the practitioner* (pp. 8-14). New York, NY: ACM.

Sahay, S., Nicholson, B., & Krishna, S. (2003). *Global IT outsourcing: software development across borders*. Cambridge, UK: Cambridge University Press.

Sawyer, S. (2004). Software Development Teams. *Communications of the ACM*, 47(12), 95-99.

Serrano, M., & Montes de Oca, C. (2004). Using the Team Software Process in an Outsourcing Environment. *Crosstalk*, 17(3), 9-13.

Smite, D. (2006). Requirements Management in Distributed Projects. Journal of Universal Knowledge Management, 1(2), pp. 69-76.

Smite, D., Moe, N. B., & Torkar, R. (2008). Pitfalls in Remote Team Coordination: Lessons Learned From a Case Study. *Lecture Notes in Computer Science*, 5089, 345-359.

Spinellis, D. (2006). Global software development in the freeBSD project. *Proceedings of the 2006 international Workshop on Global Software Development For the Practitioner* (pp. 73-79). New York, NY: ACM.

Solomon, C. M. (1995). Global Teams: the ultimate collaboration. *Personnel Journal*, 74(9), 49-58.

Sommerville, I., & Rodden, T. (1996). Human social and organizational influences on the software process, In A. Fuggetta, A. Wolf (Eds.) *Software Process. (Trends in Software, 4)*. (pp. 89-110). New York, NJ: John Wiley & Sons.

Straus, S., & McGrath, J. E. (1994). Does the medium matter? The interaction of task type and technology on group performance and member reactions. *Journal of Applied Psychology*, 79(1), 87-97.

Taweel, A., & Brereton, P. (2006). Modelling software development across time zones. *Information* *and Software Technology*, 48(1), 1-11.

Tiwana, A., Bush, A. A., Tsuji, H., Yoshida, K., & Sakurai, A. (2008). Myths and paradoxes in Japanese IT offshoring. *Communications of the ACM*, 51(10), 141-145.

Tomei, L. A. (2007). *Integrating information & communications technologies into the classroom*. Hershey, PA: Idea Group Inc.

Trigo, A., Varajão, J., Soto-Acosta, P., Barroso, J., Molina-Castillo, F.J., & Gonzalvez-Gallego, N. (2010). IT Professionals: An Iberian Snapshot. *International Journal of Human Capital and Information Technology Professionals*, *1*(1), 61-75.

Walther, J. B., & Burgoon, J. (1992). Relational communication in computer- mediated interaction. *Human Communication Research*, 19(1), 50-88.

Walz, D.B., Elam, J.J., & Curtis, B. (1993). Inside a Software Design Team: Knowledge Acquisition, Sharing, and Integration. *Communications of the ACM*, 36(10), 63-77.

Wybo, M. (2007). The IT sales cycle as a source of context in IS implementation theory, *Information & Management*, 44(4), 397-407.

Nichols, W., Carleton, A., Humphrey, Watts, & Over, J. (2009). A Distributed Multi-Company Software Project. *CrossTalk*, 22(4), 20-24.

Zanoni, R., & Audy, J. (2004). Project management model: Proposal for performance in a physically distributed software development environment. *Engineering Management Journal*, 16(2), 28-34.

Zigurs, I. (2003). Leadership in Virtual Teams:: Oxymoron or Opportunity? Organizational Dynamics, 31(4), 339-351.

Adrián Hernández López is a PhD student at the Computer Science Department of the Universidad Carlos III de Madrid in Spain. His research interests include applied research in People in IT, Human Aspects in IT and Software Process Improvement. He finished his Bachelor's degree in Computer Science on 2007 and his Master's degree in IT Science specialized on Software Engineering on 2009 at the Universidad Carlos III de Madrid. He has been working as software engineer in several companies including Telefónica and INDRA.

Ricardo Colomo-Palacios is an Associate Professor at the Computer Science Department of the Universidad Carlos III de Madrid. His research interests include applied research in Information Systems, Software Project Management, People in Software Projects and Social and Semantic Web. He received his PhD in Computer Science from the Universidad Politécnica of Madrid (2005). He also holds a MBA from the Instituto de Empresa (2002). He has been working as software engineer, project manager and software engineering consultant in several companies including Spanish IT leader INDRA. He is also an Editorial Board Member and Associate Editor for several international journals and conferences and Editor in Chief of International Journal of Human Capital and Information Technology Professionals.

Angel García-Crespo is the Head of the SofLab Group at the Computer Science Department in the Universidad Carlos III de Madrid and the Head of the Institute for promotion of Innovation Pedro Juan de Lastanosa. He holds a PhD in Industrial Engineering from the Universidad Politécnica de Madrid (Award from the Instituto J.A. Artigas to the best thesis) and received an Executive MBA from the Instituto de Empresa. Professor García-Crespo has led and actively contributed to large European Projects of the FP V and VI, and also in many business cooperations. He is the author of more than a hundred publications in conferences, journals and books, both Spanish and international.

Pedro Soto-Acosta is a Professor of Management at the University of Murcia (Spain). He holds a PhD in Management Information Systems (MISs) and a Master's degree in Technology Management from the University of Murcia. He received his BA in Accounting and Finance from the Manchester Metropolitan University (UK) and his BA in Business Administration from the University of Murcia. He attended Postgraduate Courses at Harvard University (USA). His work has been published in journals such as the European Journal of Information Systems, the International Journal of Information Management, the Information Systems Management, and the Journal of Enterprise Information Management, among others.