A study on the impact of global software development in packaged software release planning

Ricardo Colomo-Palacios
(Universidad Carlos III de Madrid, Leganés (Madrid), Spain
ricardo.colomo@uc3m.es)

Pedro Soto-Acosta
(Universidad de Murcia, Murcia, Spain
psoto@um.es)

Francisco J. García-Peñalvo
(Universidad de Salamanca, Salamanca, Spain
fgarcia@usal.es)

Ángel García-Crespo
(Universidad Carlos III de Madrid, Leganés (Madrid), Spain
angel.garcia@uc3m.es)

Abstract: Today's globalization of software development has its advantages, but also its drawbacks. Software project managers often lead the production of new software versions and their release on the market. This paper analyses the main challenges faced by software product managers in release planning with regard to the adoption of Global Software Development (GSD) practices for developing packaged software. To achieve this objective, two qualitative techniques are used in this study, namely, Focus Group and Delphi Study. The experiment produced two lists, ranking challenges in software release planning. One list was made considering the adoption of GSD practices and the other did not take into account the adoption of these practices. Results show that there are some, apparently solved, challenges for packaged software release planning like "Project monitoring & control" or "Quality management" that become crucial when facing GSD scenarios, while there are other important challenges in traditional software release planning such as "Requirements prioritisation" and "Stakeholders Management" that apparently do not add extra pressure in GSD environments. In sum, GSD is found to be highly influenced by issues concerning personnel and human resources management.

Keywords: Global Software Development; Product Manager; Release Planning, Focus Group, Delphi Study
Categories: D.2.9,

1 Introduction

Today, globalization has a major impact on the development of software. This is causing software development teams to evolve from a single site to a multiple localization working environment [Hernández-López, 10]. Software development is becoming a multi-site, multicultural, globally distributed process [Prikladnicki, 03], [García-Crespo, 10]. As a result, a new field called Global Software Development

(GSD) has emerged to cover specific aspects of global distributed software development [Oshri, 07]. Not in vain are GSD and software development outsourcing integral parts of software projects [Schümmer, 09] and, because of their multi-site location, global resources are becoming pervasive in the software industry [Ramasubbu, 05]. In this scenario, firms developing or maintaining software products cannot ignore the impact of GSD [Cusick, 06], since it is driving a deep transformation in the way that products are conceived, designed, constructed, tested, and delivered to customers.

GSD teams are geographically distributed teams which make use of collaborative technologies to produce software [Herbsleb, 01]. These teams can be considered as a specification of virtual teams [Martins, 04] and their creation is encouraged by the relations between customers of software development outsourcing organizations and developers [Heeks, 01]. Several factors justify the adoption of GSD within the IT industry nowadays [Herbsleb, 01], for instance:

- proximity to business markets, which facilitates a better knowledge of customers and local conditions;
- increasing pressure to improve time-to-market, which can be achieved by using time-zone differences in 'round-the-clock' development;
- the need to have a global resource pool and so have successful, cost-competitive resources, wherever located.

Although using GSD teams can be very productive, they do suffer from three types of distance: geographical, temporal and socio-cultural [Conchuir, 09]. The literature suggests that these distances may affect three important aspects in software development: communication, coordination and control. Moreover, the literature suggests that other difficulties may arise from this shift in software development. These are reported in Table 1:

Difficulty	**Literature support**
Communication, coordination, and control	[Battin, 01], [Conchuir, 09], [Cramton, 02], [Cusumano, 08], [Herbsleb, 99], [Herbsleb, 03], [Hinds, 03], [Kommeren, 07], [Kotlarsky, 05], [Kraut, 95], [Krishna, 04], [Layman, 06], [MacDuffie, 08], [Marquardt, 01], [Metiu, 06], [Olson, 00], [Prikladnicki, 03], [Sooraj, 08], [Taxén, 06]
problems of knowledge transfer	[Chua, 06], [Conchuir, 09], [Mattarelli, 09], [Prikladnicki, 03]
Issues regarding the protection of intellectual property	[Herbsleb, 01], [Sakthivel, 07]
Less efficiency	[Herbsleb, 01], [Kommeren, 07], [Milewski, 09], [Rogers, 05]
Higher conflict rates	[Herbsleb, 03]
Disparities in team members' strategies, behaviour and assumptions about the work at hand and how to work with others	[Carmel, 99], [Cusumano, 08], [Goodman, 91], [Kotlarsky, 05], [MacDuffie, 08], [Mattarelli, 09], [Metiu, 06], [Milewski, 09], [Olson, 00], [Prikladnicki, 03]

Differences in opinion about the nature of the software development process	[Cusumano, 08], [Nicholson, 01], [Olson, 00]
High failure rates	[Fabriek, 08]
Lack of trust	[Barczak, 06], [Das, 98], [Derosa, 04], [Gorton, 96], [Hernández-López, 10], [Jarvenpaa, 98], [Olson, 00], [Oza, 06]
Lack of Quality	[Seshagiri, 06]
Socio-Cultural distance	[Ali Barbar, 07], [Damian, 03], [Ebert, 01], [Evaristo, 03], [Gorton,96], [Krishna, 04], [Layman, 06], [Marquardt, 01], [Mortensen, 01], [Nicholson, 01], [Prikladnicki, 03]

Table 2: GSD problems

A recent and extensive review of the challenges faced by GSD can be found in Jimenez [2009]. However, in spite of these difficulties, GSD provides several outstanding benefits, which have been reported in the literature (see table 2).

Benefit	**Literature support**
Greater availability of human resources and multi-skilled workforce	[Carmel, 01], [Carmel, 05], [Conchuir, 09], [Ebert, 01], [Herbsleb, 01], [Jalote, 06], [Jiménez, 09], [Kommeren, 07], [Milewski, 09], [Suzuki, 99]
Lower Costs	[Conchuir, 09], [Jiménez, 09], [Kobitzsch, 01], [Kommeren, 07], [Ramasubbu, 05], [Sooraj, 08]
Strategic regional presence for improved customer service	[Ramasubbu, 05]
shorter time-to-market cycles	[Carmel, 99], [Herbsleb, 01], [Jalote, 06], [Kommeren, 07], [Sooraj, 08]
improvement in the ability to respond quickly to local customer needs	[Herbsleb, 01], [Kommeren, 07]
The mix of developers with different cultural backgrounds may foster new ideas	[Carmel, 05], [Conchuir, 09], [Ebert, 01], [Highsmith, 01], [Kommeren, 07]
Productivity improvements	[Kesner, 07]
Efficiency	[Conchuir, 09], [Kommeren, 07]
Access to new markets	[Karolak, 98]
improved documentation	[Conchuir, 09], [Gumm, 06]

Table 2: GSD benefits

As can be derived from Table 1 and Table 2, the globalization of software development introduces a great deal of complexity in an already complex process

[Treinen, 06]. Thus, working in a global context has its advantages, but also its drawbacks [Ebert, 01].

There are several ways to organize project works under GSD. In some cases, teams' work is organized in a sequential manner, such as in the case of the 24-Hour Knowledge Factory [Gupta, 2007]. This structure allows the execution of tasks with faster turnaround time, which has been suggested as one of the major potential benefits of distributing work across time zones [Gupta, 2009], although it has its drawbacks as well. Another way of collaboration is parallel development [Ebert, 01]. This approach, which benefits from software modularization, reduces the communication needs between development sites which, in turn, sometimes leads to problems at the integration stage [Conchuir, 09]. According to [Akmanligil, 04], there are many different implementation and design strategies. One option includes the development of various phases locally (requirements gathering and the construction of the various subsystems) and, then, connect them through bridges. Another variation occurs when representatives of teams at different locations gather local requirements, then, come together at a central site, try to resolve the differences and, thus, define a common structure. Another possibility is that the project is broken into multiple components in the beginning and, then, different components are designed and built at distinct locations.

Another important issue is related to the shift in the software market, from developing customized software to primarily developing software as a standard product [Xu, 2007]. Within software products, packaged software has emerged as a key to gain competitive advantages in an ultra competitive market. However, most organisations devoted to the development of software packages face the problem of including GSD practices in their software product evolution in order to benefit from their intrinsic advantages. This is the focus of this study. Thus, taking into account the importance and influence of GSD in packaged software release planning, this paper analyses the main challenges faced by software product managers in release planning with regard to the adoption of GSD for developing packaged software.

The paper consists of four sections and is structured as follows. Section 2 reviews the relevant literature about GSD, software product managers and release planning. Section 3 describes the study conducted with a sample of software product managers about the influence and challenges of packaged software release planning in GSD environments. Finally, the paper ends with a discussion of research findings, limitations and concluding remarks.

2 Literature Review

We live in an era in which we can no longer afford—in terms of either time or money—to custom-develop every system [Ncube, 08]. Thus, in the past few years, the demand for computer software packages has increased rapidly among firms. Software companies have detected this need and have developed a variety of packages in response to this demand [Jadhav, 09]. However, this response varies from one software company to another with regard to the software product developed. In the literature, the boundaries distinguishing shrink-wrapped software, commercial off the-shelf software (COTS), packaged and commercial software are blurred, but the principle of 'Make one, sell many' is common to them all [Xu, 07]. In spite of their

differences, [Sawyer, 00] considers packaged software as all software sold as a tradable product (purchased from a vendor, distributor or store) for all computer platforms. Our study adopts the definition of Packaged Software proposed by [Xu, 07], who defined packaged software as ready-made software product that can be readily obtained from software vendors and which generally require little modification or customization. According to the definition, the term typically refers to upscale enterprise software suites, such as enterprise resource planning (ERP) or customer relationship management (CRM) systems.

The creation/design of a packaged software product is typically driven by the vision of a small group - perhaps even one person -, who is indispensable to its production. The literature recognizes that from these key individuals many innovations arise that define a product [Carmel, 98]. Initially, it is expected that packaged software developers work better when they share common spaces [Sawyer, 97]. However, this is not the case because software is developed collaboratively in multiple locations around the world, and projects are being contracted out in whole or in part [Madachy, 08].

In these circumstances, traditional software developers of packaged software must adapt not only to the new demands but also to the new working sets. One traditional and important figure in the packaged software development process is the product manager. According to [Ebert, 07], the success of any product depends on the skills and competences of its product manager. However, although the role played by the software product manager has been extensively addressed in the literature (e.g. [Barney, 08]; [Ebert, 2008]; [Fricker, 10]; [Karlsson, 07]; [Lehtola, 06]; [Trienekens, 09]; [van de Weerd, 10]; [Van den Akker, 08]; [Wallin, 02]), his or her role in software engineering has not been summarized in a comprehensive perspective so far [Ebert, 07].

The product manager is responsible, among other things, for product definition, product release and product lifecycles, creating an effective multifunctional product introduction team and, above all, preparing and implementing the business case [Ebert, 07)]. The product management practice contains activities that are carried out on operational, tactical, and strategic levels [van de Weerd, 10]. According to [van de Weerd, 06], there are four process areas within software product management: requirements management, release planning, product roadmapping, and portfolio management. The Activities included in the first two areas are mainly on an operational level, whereas the latter ones contain tactical and strategic activities [van de Weerd, 10]. In this sense, before starting a project, the product manager presents the idea to managers from different management areas such as development, marketing and sales, service and maintenance, manufacturing, training, and so on, who together decide whether to start the project [Wallin, 02]. In fact, product managers are deeply involved with requirements engineering. That is, product managers obtain functional, technical and usability requirements from distinct sources such as business departments (sales, customer service, maintenance, development...), ad hoc or organized customer contacts, user groups and so on [Natt och Dag, 2005]. Nonetheless, their role is also important in other stages of the development, such as release planning [van den Akker, 08]. Project managers often lead the production of new software versions, new software-related products and their release on the market.

They are, thus, responsible for many of the innovations that define a software product [Carmel, 98].

Human resources management are key in software development processes [Chang, 10], [Colomo-Palacios, 10], [Naranjo-Gil, 09], [Trigo, 10]. This is also important for the product manager, who needs to manage both the soft and technical skills of his/her personnel. Although several works in the literature consider that the role of the product manager is linked to GSD ([Conchuir, 09], [Ebert, 08], [Nicholson, 01], [Oshri, 07], [Prikladnicki, 09], [Regnell, 01], [van de Weerd, 10]), none have studied nor discussed the influence that GSD has on one of the product managers' main responsibilities: Release Planning.

The process of product release planning addresses decisions related to the selection and assignment of features to create a sequence of consecutive product releases that satisfy certain constraints such as technical resources available, budget and risk borne [Ruhe, 05]. A major problem faced by companies developing or maintaining large and complex systems has to do with determining which features, normally from a large set, should be assigned to each software product release. In addition, there is the question of how to assign resources accordingly [Ngo-The, 09].

Release planning can be done informally or in a more formalized way [Momoh, 06. This is the difference between the art of release planning and the science of release planning. The first approach relies on human intuition, communication, and capabilities to negotiate between conflicting objectives and constraints, while the latter formalizes the problem and applies computational algorithms to generate the best solutions [Ruhe, 05]. The work of [Svahnberg, 10] provides a systematic review of the release planning approaches.

Whether using a formal or an informal way, software vendors and, more precisely, software product managers tackle release planning with the added difficulty of handling issues regarding GSD. Although there is research that deals with issues related to release planning in GSD teams (e.g. [Damian, 07], [Layman, 2006], [van de Weerd, 10]), there is a need to investigate further into the software product managers´ viewpoint with regard to the influence of GSD on software product release planning.

3 Study: The influence of GSD in Packaged Software Release Planning

3.1 Research methodology

The shift in focus of information systems research, from technological to managerial and organizational issues, has made qualitative research methods increasingly useful [Myers, 97]. Thus, the study of the influence of GSD on Release Planning was carried out here by conducting a qualitative research approach based on two techniques: Focus Group and Delphi Study. These research methods are very useful when the purpose is to explore an area of interest, obtain an overview of a complex area and/or discover differences rather than similarities. As a consequence of both the importance of GSD and the suitability of qualitative research in such environments, the use of this methodology is widespread among GSD studies (e.g. [Barcus, 08], [Espinosa, 07], [García-Crespo, 10], [Kotlarsky, 05], [Kotlarsky, 08], [Layman, 2006], [Oshri, 2007], [Pauleen, 04]).

The aim of our study is to explore the nature and importance of the challenges found, when software products release planning is developed through GSD.

The study consists of two steps, and each is divided into three consecutive phases: planning, data collection and analysis (see Figure 1).

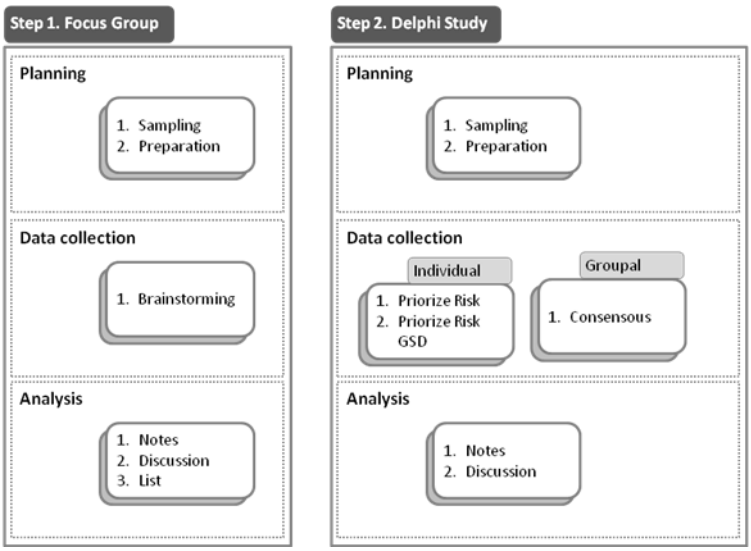

Figure 1: Research steps and phases

As depicted in Figure 1, two qualitative techniques are used in this study, namely, Focus Group and Delphi Study. Focus groups involve assembling small groups of peers to discuss particular topics [Baddoo, 02]. Discussion within these groups, although directed by a researcher, is largely free-flowing [Hall, 02]. The use of discussion groups in software engineering and information systems development research activities has been extensively treated in the literature (e.g. [Baddoo, 02], [Beecham, 05], [Benett, 08], [Casey, 10], [Christensen, 10], [Hall, 02], [Karlsson, 07], [Runeson, 06]).

The Delphi method is likewise equally relevant for qualitative research. The Delphi method owes its name to the ancient Greek oracle at Delphi. The oracle used to offer visions of the future to those who sought advice [Cassino, 84]. The Delphi technique is generally considered to be an appropriate method for studies that lack historical data and require the collection of expert opinions [Gallego, 07]. It is based on a survey with experts that presents three features ([Landeta, 2006]; [Hsu, 10]): anonymous response; iteration and controlled feedback; and statistical group response. This method has proven to be a popular and effective tool in recent software engineering and information systems research (e.g. [Bañuls, 08]; [García-Crespo, 10]; [Iden, 10]; [Kasi, 08]; [Liu, 10]; [Nakatsu, 09]).

As stated above the experiment consisted of two steps. In Step 1, a panel of experts (software product managers) was called together to identify the main challenges regarding the release planning of software products. Once all the challenges had been identified, in Step 2, by means of a Delphi study, a panel of

software product managers prioritized the challenges. Two lists resulted from the experiment. The first list ranked challenges according to their risk for the whole project, while the second list ranked challenges with respect to the risk added as a consequence of the adoption of a GSD approach.

3.2 Step 1: Focus Group

PLANNING

The aim of the focus group was to obtain a list of challenges for software product release planning. To achieve this objective, a group of five software product managers from 3 different companies was selected. The sample consisted of one woman (20%) and four men (80%), with an average age of 39.7. Subjects were selected from those who answered positively to a personal invitation sent to a set of companies related to the authors.

DATA COLLECTION

The meeting was designed to be attended by three researchers (one in each location). Participants were connected using videoconference and assisted on-site by the researcher. The focus group's virtual meeting lasted approximately 50 minutes. During the meeting, researchers took extensive notes.

The task consisted of identifying the main challenges for packaged software release planning. In accordance with previous literature [Karlsson, 07], the session started with a brainstorming, where subjects thought about challenges from their past experience and wrote them down on post-it notes. Once this step was completed, they discussed for 30 minutes the importance of each challenge and chose the final list.

ANALYSIS & RESULTS

This subsection presents the challenges raised during the meeting of the focus group. The analysis was carried out using NVIVO 2.0 (International QSR Pty Ltd), a software for qualitative data analysis. Table 3 lists in alphabetical order the challenges encountered and explained by using excerpts from direct transcripts of the focus group session.

Challenge	**Excerpts**
Effort and time estimate	"It is never easy to estimate the effort needed when developing software" "The initial estimate is almost a fairy tale sometimes" "There is like a cascade of new requirements that makes difficult to give a good time estimate" "We are always re-planning"
Features & releases scheduling	"It's very difficult to establish a project's scope" "Stakeholders want all features as soon as possible" "It's very difficult to assign values to releases and, because of this, compare them" "Technological and resource constraints are not taken into account in release

	scheduling, only business value is considered"
Project monitoring & control	"Outsourcing the development leads to our having resource reports only occasionally" "We have problems with information flow, and this has an impact on project monitoring" "In many cases, outsourcing decisions do not flow fluently and, thus, some people are not aware of the state of the project"
Quality management	"Ultra-quick releases are difficult to control in terms of quality" "Outsourced developments have a quality process separated from ours"
Requirements prioritisation	"Although we have a requirements prioritisation assignment policy, it's only a reference" "It's very hard to know the exact interrelationships of requirements" "We usually discover new requirements as we evaluate others"
Staffing	"It is always difficult to get the human resources you need" "Internal selection is a nightmare in terms of resource availability" "The problem is sometimes worse when you go outside" "Many times, the initial project staff has nothing to do with the final team" "Key people are always busy doing other things" "We have a big turnover in several projects"
Stakeholder management	"We cannot reach a good level of involvement from stakeholders in release planning" "In many cases, stakeholders take long time to vote candidate features" "Stakeholders are reluctant to discuss or negotiate their priorities" "Stakeholders are globally distributed and it's not easy to get them all together" "Stakeholders have not a business vision"
Task assignment according to resource/team competences	"Despite having a record of resource competences, in many cases, it's not useful"

	"Unfortunately, task assignment is, in many cases, FIFO or LIFO, while in others SJF or even no resource-competence criteria are used" "When assigning work packages, many project managers fight for the shortest or the longest work package, without taking into account the content" "When we hire the development outside, we never know the real competence of the partner and, hence, the distribution of the tasks is not performed in a right way" "We are not sure about the skills required and roles needed"

Table 2: Challenges discovered in focus group meeting

A quick look at the challenges encountered confirms the importance of many aspects previously reported in the literature on release planning (e.g. [Carlshamre, 02], [Greer, 09], [Momoh, 06], [Ruhe, 05], [Saliu, 05]). However, it is important to notice that GSD or related approaches are mentioned in four challenges: Project monitoring & control; Quality management; Staffing; and Task assignment according to resource/team competences. This is not a trivial issue since, as we try to confirm with this work, GSD approach has nowadays a deep impact in packaged software release planning. The output of this phase consisted of a list of challenges that were then ranked by a Delphi study.

3.3 Step 2: Delphi Study

PLANNING

The objective of the Delphi study presented in Step 2 was twofold. First, to rank the packaged software release planning challenges and, second, to rank these challenges with respect to the risk added as a consequence of the adoption of a GSD approach. To achieve this objective, eighteen software product managers from ten different companies were selected on the basis of their experience of using GSD. Since the literature recommends a Delphi panel size ranging from 10 to 18 experts [Okoli, 04], the composition of the sample was considered ideal. Moreover, regarding the selection of experts, the selection was made following [Delbecq's, 75] guidelines. Thus, we believe that experts had significant knowledge about the problem studied. None of the subjects participated in Step 1 (Focus Group) and Step 2 (Delphi Study). The demographic composition of the sample reveals that 5 of the participants were female (28%) while 13 were male (72%); the average age of the sample was 41.6 years. Subjects were selected from those who answered positively to a personal invitation sent to a set of companies related to the authors.

DATA COLLECTION

First, panellists were asked to rank the release planning challenges obtained in step 1 according to the two criteria described in the previous section. In the first round, subjects performed the two rankings individually assisted by one researcher (by

phone, videoconferencing or in person). Panellists took 42 minutes on average to complete this initial round. Once all the records had been collected, two overall rankings were computed based on individual answers. These two rankings were later presented to subjects in Delphi round 2. The task in round 2 consisted of creating an agreed group response for both rankings.

The consensus of the first round can be described by the Kendall coefficient of concordance (W). W measures the degree of association among k sets of rankings. Kendall's W has a value between 0 and 1. [Schmidt, 01] proposed that strong consensus exists for W >= 0.7; moderate consensus for W = 0.5; and weak consensus for W < 0.3.

ANALYSIS & RESULTS

The results from the first round are presented in Figure 2. The challenges identified are shown in columns, while cylinders represent the score given by the panellists to each challenge of software release planning:

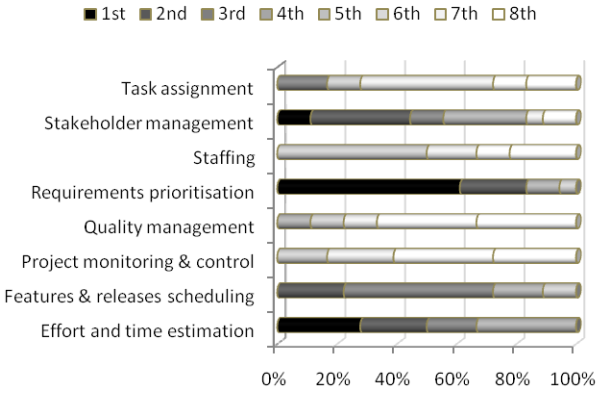

Figure 2: Scores for release planning challenges

Figure 3 depicts the ranking of release planning challenges when considering GSD specific problems.

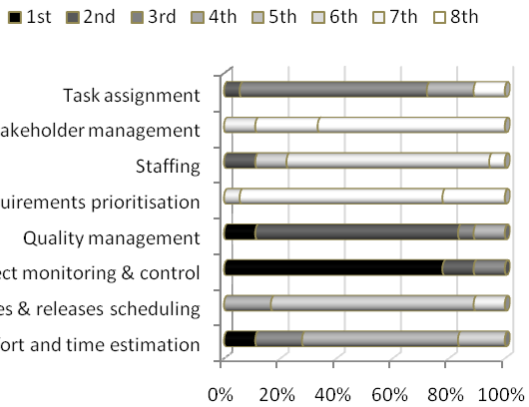

Figure 3: Scores for release planning challenges when the process is supported by GSD

A quick look at figures 2 and 3 suggests that figure 3 scores are much more concentrated and, hence, more degree of consensus may exist. In fact, the Kendall coefficient of concordance test confirms these differences, with $W = 0.676$ ($n = 18$, $P < 0.01$) for the first ranking task and $W = 0.822$ ($n = 18$, $P < 0.01$) for the second one. These figures corroborate that, although the agreement among subjects regarding the first ranking is moderate-high, the agreement with regard to the second ranking is remarkable.

Once all data had been gathered and analyzed, researchers prepared two rankings for the two research objectives based on the panellists' rankings. Table 3 shows both rankings as the final output for Delphi round 1:

General Challenges Ranking	**Specific GSD Challenges Ranking**
1. Requirements prioritisation	1. Project monitoring & control
2. Effort and time estimate	2. Quality management
3. Features & releases scheduling	3. Effort and time estimate
4. Stakeholder management	4. Task assignment
5. Task assignment	5. Features & releases scheduling
6. Staffing	6. Staffing
7. Quality management	7. Requirements prioritisation
8. Project monitoring & control	8. Stakeholder management

Table 3: Round 1, general and GSD specific challenge rankings

Panellists were asked to rank both lists in the second round. The input of the process consisted of the two lists generated in round one, while the output comprised two lists that represented the agreement among panellists about general and Specific GSD challenges in software release planning. The final rankings obtained are listed in Table 4.

General Challenges Ranking	**Specific GSD Challenges Ranking**
1. Requirements prioritisation	1. Project monitoring & control
2. Stakeholder management	2. Quality management
3. Features & releases scheduling	3. Effort and time estimate
4. Effort and time estimate	4. Task assignment
5. Task assignment	5. Features & releases scheduling
6. Staffing	6. Staffing
7. Quality management	7. Requirements prioritisation
8. Project monitoring & control	8. Stakeholder management

Table 4: Round 2, general and GSD specific challenge rankings

Round 2 reveals three important findings. The first is that GSD challenges ranking is the same as that obtained in Round 1, although this result may not be surprising because of the high level of agreement obtained in phase 1 ($W = 0.822$). The second finding refers to the importance that panellists give to one general challenge (Stakeholders Management). Actually, this challenge went from fourth

position to second in phase 2, swapping places with the "Effort and time estimate" challenge. Finally, it is important to point out that the first two challenges in the general ranking "Requirements prioritisation" and "Stakeholders Management" were the last ones in the GSD challenges ranking, swapping their positions with "Quality management" and "Project monitoring & control".

3.4 Discussion

When focusing on GSD challenges and their relation with general challenges, there are several aspects to be noted. First, authors highlight the lack of importance of requirements prioritization in a GSD scenario in comparison to the traditional ones. The interaction of requirements engineering and GSD has been a subject of recent research (e.g., [Bhat, 06]; [Damian & Zowghi, 03]; [Damian, 07]; [Port, 09]). Although it has been pointed out that GSD are an attractive and promising research area in GSD (Herbsleb, 2007), the similarity of the process to the traditional development implies that aspects such as prioritization and negotiation are not considered as distinguishing elements in relation to the traditional process as [García-Crespo, 10] pointed out in a recent work. This argument can also be applied to Stakeholder Management. Works devoted to this issue are relevant and numerous, with the work of [Damian, 07] being, perhaps, that of deepest influence. However, these studies are related to stakeholder management of GSD stakeholders rather than the implications of GSD for traditional stakeholders.

In spite of this, it is important to point out that both "Requirements prioritization" and "Stakeholder Management" are, in a sense, connected. Not in vain, according to [Greer, 04], is one of the challenges to the software engineering research community to involve stakeholders in the requirements engineering process. Taking this into account, it is logical to find that both challenges are quite near in both rankings.

The second aspect is the importance of "Project monitoring & control" and "Quality management" for software product managers with respect to the implications for GSD environments. Focusing on the first challenge, indeed the top ranked challenge in the GSD ranking, GSD has been pointed to as a factor affecting project control activities [Damian et al., 03]. Moreover, control in distributed environments is even more challenging and requires specific project management and reporting mechanisms [Ralité, 08]. This may be because the management of global software development is a difficult and complex task [Ebert, 01] - a more difficult and complex task to undertake than with collocated projects [Oshri, 08]. According to [Casey, 10], it is clear that the management of GSD teams requires the role and responsibilities of the project manager to be extended. To be effective, the new management strategy must include monitoring and controlling the additional variables, social and technical, caused by operating in a GSD environment: communication, cooperation, coordination and visibility [Casey, 09]. This new scope requires new competences and management skills that, in fact, make software project management even more difficult.

With respect to "Quality management", according to [Jiménez, 09], the quality of products developed in GSD environments is highly influenced by the quality of the processes that support them. With a focus similar to that of other well known models like CMMi, ITSqc, the eSourcing Capability Model (eSCM) was designed at Carnegie Mellon University in order to improve the relationship between IT Services

providers and their customers. eSCM is two-fold: eSCM-CL for Clients [Hefley, 06] and eSCM-SP for Service Providers [Hyder, 04]. In a latter work, [Siakas, 06] discussed the implications of GSD and presented a model, namely Software Quality Management—Cultural and Organisational Diversity Evaluation (SQM-CODE) to bring added value for service purchasers in their search for, selection of and collaboration with service providers. Thus, although there are paths described in the literature, the inclusion of such new processes implies a higher level of complexity to software processes.

Regarding "Effort and time estimate", estimation is more problematic in distributed environments as a result of volatile requirements, changing specifications, cultural diversity, and the lack of informal communication [Jiménez, 09]. Due to these new factors, effort estimating methods are lacking in accuracy [Conchúir, 09]. In a recent work, [Muhairat, 10] argues that the development of a software product in a globally distributed environment consumes more effort and more time to complete than the estimating tools suggest. This lack of accuracy was reported by the panelists and latter introduced as the third factor in importance for GSD related challenges.

Finally, concerning "Task assignment", this challenge appears in every software development project. Not in vain does resource allocation in software development projects constitute an extremely hard problem, and it is one of the principal challenges of software project management due to its sheer complexity [Chang, 01]. On the one hand, properly assigning people to development roles is crucial for creating productive software development teams [Acuña, 04]. On the other hand, wrong assignments may result in significant loss of value due to understaffing, under-qualification or over-qualification of assigned personnel and high turnover of poorly matched workers [Naveh, 07]. In a GSD scenario, like that reported by [García-Crespo et al., 10], this issue relates mainly to the necessity of preserving core competency in a distributed and increasingly competitive environment. Thus, it is sometimes difficult for managers to combine core competency preservation with time and schedule pressure.

To sum up, there are some, apparently solved, issues for software development like "Project monitoring & control" or "Quality management" that become crucial challenges when facing GSD scenarios. There are also key issues for software release planning such as "Requirements prioritisation" and "Stakeholders Management" that apparently do not add extra pressure in GSD environments. Therefore, we can conclude that in packaged software release planning, GSD brings out more complexity only to non-specific release activities.

3.5 Limitations

The main limitation of our study is the sample size. Although it is an introductory investigation, it uses a small sample size and, thus, conclusions and implications may not be broadly generalized. Future studies could include the total population of the target group or the whole industry.

Another limitation comes from the fact that all participants were Europe. This limitation could be also a restriction in the legitimacy of the conclusions taking, since today software is being developed all over the world and outsourced to almost any place.

4 Conclusions and Future Work

GSD inject more complexity into the already complex process of software release planning. The pressure of reducing time-to-market ratios of product releases and cutting costs is pushing software vendors to adopt GSD to benefit from the its inherent advantages. However, this new working approach brings both benefits and risks. From the software product manager's perspective, this paper aims to identify and rank these challenges. Results from a qualitative study show that issues such as the lack of transparency in resource and team competences, the lack of effective reporting or the absence or the lack of uniformity of the offshore company's quality process are important challenges when implementing GSD.

According to the subjects' ranking, there is a set of activities or challenges that are already complex without adopting GSD: Stakeholder management or Requirements prioritisation, and others that are more difficult in this new scenario, like Task assignment according to resource/team competence, Project monitoring & control or Quality Management. However, it's important to point out that GSD complexity is related to non-specific software release planning activities.

Considering the first challenge of the second group, "Task assignment according to resource/team competence", we can conclude that people are still a resource of paramount importance for software development, no matter how or where it is developed. After all, software is still being developed by people. Human resources are crucial in packaged software, no matter how it is developed. To address these issues in the future, three different research lines are proposed. The first is the design of a method for knowledge extraction and expert location for global projects based on individual and group competence. The second is the design of a method capable of integrating competence-oriented work package assignment and time and resource constraints related to release planning. The third line of research proposed is aimed at designing a method to assess the performance of the GSD service provider in terms of business value and contribution to the overall success of the project.

Acknowlegments

We would like to thank *Fundación CajaMurcia* for the financial support provided.

References

[Acuña, 04] Acuña, S.T., & Juristo, N., Assigning people to roles in software projects, Software—Practice and Experience, 34(7), 2004, pp. 675–696.

[Akmanligil, 04] Akmanligil, M., & Palvia, P.C., Strategies for global information systems development, Information & Management, 42(1), 2004, pp. 45-59.

[Ali Barbar, 07] Ali Barbar, M., Verner, J. M., & Nguyen, P. T., Establishing and maintaining trust in software outsourcing relationships: An empirical investigation, Journal of Systems and Software, 80 (9), 2007, pp. 1438-1449.

[Baddoo, 02] Baddoo, N., & Hall, T., Motivators of Software Process Improvement: an analysis of practitioners' views, The Journal of Systems and Software, 62, 2002, pp. 85-96.

[Bañuls, 08] Bañuls, V. A., & Salmeron, J. L., Foresighting key areas in the Information Technology industry, Technovation, 28 (3), 2008, pp. 103-111.

[Barcus, 08] Barcus, A., & Montibeller, G., Supporting the allocation of software development work in distributed teams with multi-criteria decision analysis, Omega, 36 (3), 2008, pp. 464-475.

[Barczak, 06] Barczak, G., McDonough, E., & Athanassiou, N., So you want to be a global project leader?, Research Technology Management, 49 (3), 2006, pp. 28-35.

[Barney, 08] Barney, S., Aurum, A., & Wohlin, C., A product management challenge: Creating software product value through requirements selection, Journal of Systems Architecture, 54 (6), 2008, pp. 576-593.

[Battin, 01] Battin, R. D., Crocker, R., Kreidler, J., Subramanian, K., Leveraging resources in global software development, IEEE Software, 18 (2), 2001, pp. 70-77.

[Beecham, 05] Beecham, S., Hall, T., & Rainer, A., Defining a Requirements Process Improvement Model, Software Quality Journal, 13 (3), 2005, pp. 247-279

[Benett, 08] Bennett, C., Myers, D., Storey, M.-A., German, D. M., Ouellet, D., Salois M., & Charland, P., A survey and evaluation of tool features for understanding reverse engineered sequence diagrams, Journal of Software Maintenance and Evolution: Research and Practice, 20 (4), 2008, pp. 291-315.

[Bhat, 06] Bhat, J.M., Gupta, M., & Murthy, S.N., Overcoming Requirements Engineering Challenges: Lessons from Offshore Outsourcing, IEEE Software, 23 (5), 2006, pp. 38–44.

[Carlshamre, 02] Carlshamre, P., Release Planning in Market-Driven Software Product Development: Provoking an Understanding, Requirements Engineering, 7 (3), 1998, pp. 139-151.

[Carmel, 98] Carmel, E., & Sawyer, S., Packaged software development teams: what makes them different?, Information Technology & People, 11 (1), 1998, pp. 7-19.

[Carmel, 99] Carmel, E., Global Software Teams: Collaborating Across Borders and Time Zones, Prentice Hall, New York, 1999.

[Carmel, 01] Carmel, E, & Agarwal, R., Tactical approaches for alleviating distance in global software development, IEEE Software, 18 (2), 2001, pp. 22-29.

[Carmel, 05] Carmel, E, & Tija, P., Offshoring Information Technology: Sourcing and Outsourcing to a GlobalWorkforce, Cambridge University Press, Cambridge, 2005.

[Casey, 09] Casey, V., & Richardson, I., Implementation of global software development: a structured approach, Software Process Improvement and Practice, 14 (5), 2009, pp. 247–262.

[Casey, 10] Casey, V., Virtual software team project management, Journal of the Brazilian Computer Society, 16 (2), 2010, pp. 83-96.

[Cassino, 84] Cassino, K., Delphi panel: A practical 'Crystal Ball' for researchers, Marketing News, 18 (2), 1984, pp. 10-11.

[Chang, 01] Chang, C.K., Christensen, M.J., & Zhang, T. (2001). Genetic Algorithms for Project Management. Annals of Software Engineering, 11 (1), 2001, pp. 107–139.

[Chang, 10] Chang, C. L., The study of the turnover of MIS professionals—The gap between Taiwanese and US societies, International Journal of Information Management, 30 (4), 2010, pp. 301-314.

[Christensen, 10] Christensen, H. B., & Hansen, K. M., An empirical investigation of architectural prototyping, Journal of Systems and Software, 83 (1), 2010, pp. 133-142.

[Chua, 06] Chua, A. L., Pan, S., Knowledge transfer in offshore insourcing, Proceedings of the 27th international conference on information systems, 2006, pp 1039-1053.

[Colomo Palacios, 2010] Colomo Palacios, R., Tovar Caro, E., García Crespo, A., & Gómez Berbís, J. M., Identifying Technical Competences of IT Professionals: The Case of Software Engineers, International Journal of Human Capital and Information Technology Professionals, 1 (1), 2010, pp. 31-43.

[Conchuir, 09] Conchuir, E. O., Holmstrom-Olson, H., Agerfalk, P. J., & Fitzgerald, B., Benefits of Global Software Development: Exploring the Unexplored, Software Process Improvement and Practice, 14 (4), 2009, pp. 201-212.

[Cramton, 02] Cramton, C. D., Attribution in distributed work groups, in: P. J. Hinds, S. Kiesler, (Eds.), Distributed Work, Cambridge, MA: MIT Press,2002, pp. 91-212.

[Cusick, 06] Cusick, J., & Prasad, A., A Practical Management and Engineering Approach to Offshore Collaboration, IEEE Software, 23 (5), 2006, pp. 20-29.

[Cusumano, 08] Cusumano, M. A., Managing Software Development in Globally Distributed Teams, Communications of the ACM, 51 (2), 2008, pp. 15-17.

[Damian, 03] Damian, D., Global software development: growing opportunities, ongoing challenges, Software Process: Improvement and Practice, 8 (4), 2003, pp. 179-182.

[Damian et al., 03] Damian, D., Lanubile, F., & Oppenheimer, H. L., Addressing the challenges of software industry globalization: the workshop on global software development, in Proceedings of the 25th International Conference on Software Engineering, pp. 793–794, Portland, Ore, USA, May 2003.

[Damian & Zowghi, 03] Damian, D.E. & Zowghi, D., Requirements Engineering challenges in multi-site software development organizations, Requirements Engineering Journal, 8 (3), 2003, pp. 149–160.

[Damian, 07] Damian, D., Stakeholders in Global Requirements Engineering: Lessons Learned from Practice, IEEE Software, 24 (2), 2007, pp. 21-27.

[Das, 98] Das, T. K., & Teng, B. S., Between trust and control: developing confidence in partner cooperation in alliances, The Academy of Management Review, 23 (3), 1998, pp. 491-512.

[Delbecq, 75] Delbecq, A. L., Ven, A. H. V. D., & Gustafson, D. H., Group Techniques For Program Planning, Glenview, Scott Foresman, 1975.

[Derosa, 04] Derosa, D., Hantula, D. A., & D'Arcy, J., Trust and leadership in virtual teamwork: a media naturalness perspective, Human Resource Management, 43 (2-3), 2004, pp. 219-233.

[Ebert, 01] Ebert, C., & Neve, P. D., Surviving global software development, IEEE Software, 2001, pp. 62-69.

[Ebert, 07] Ebert, C., The impacts of software product management, Journal of Systems and Software, 80 (6), 2007, pp. 850-861.

[Ebert, 08] Ebert, C., & De Man, J., Effectively utilizing project, product and process knowledge, Information and Software Technology, 50 (6), 2008, pp. 579-594.

[Espinosa, 07] Espinosa, J. A., Slaughter, S. A., Kraut, R. E., & Herbsleb, J. D., Team Knowledge and Coordination in Geographically Distributed Software Development, Journal of Management Information Systems, 24 (1), 2007, pp. 5-12.

[Evaristo, 03] Evaristo J. R., Scudder, R., Desouza, K., & Sato, O., A dimensional analysis of geographically distributed project teams: a case study, Journal of Engineering Technology and Management, 11 (4), 2003, pp. 58-70.

[Fabriek, 08] Fabriek, M., Brand, M. van de Brinkkemper, S., Harmsen, F., & Helms, R. W., Reasons for Success and Failure in Offshore Software Development Projects, European Conference on Information Systems, 2008.

[Fricker, 2010] Fricker, S., Gorschek, T., Byman, C., & Schmidle, A., Handshaking with Implementation Proposals: Negotiating Requirements Understanding, IEEE software, 27 (2), 2010, pp. 72-80.

[Gallego, 07] Gallego, M. D., Luna, P., & Bueno, S., Designing a forecasting analysis to understand the diffusion of open source software in the year 2010, Technological Forecasting and Social Change, 75 (5), 2007, pp. 672-686.

[García-Crespo, 2010] García-Crespo, Á., Colomo-Palacios, R., Soto-Acosta, P., & Ruano-Mayoral, M., A Qualitative Study of Hard Decision Making in Managing Global Software Development Teams, Information Systems Management, 27 (3), 2010, pp. 247-252.

[Goodman, 91] Goodman, P. S., & Leyden, D. P., Familiarity and group productivity, Journal of Applied Psychology, 76 (44), 1991, pp. 578-586.

[Gorton, 96] Gorton, I., & Motwani, S., Issues in co-operative software engineering using globally distributed teams, Information and Software Technology, 38 (10) , 1996, pp. 647-655.

[Greer, 04] Greer, D., & Ruhe, G., Software release planning: an evolutionary and iterative approach, Information and Software Technology, 46 (4), 2004, pp. 243-253.

[Greer, 09] Greer, D., & Conradi, R., Software project initiation and planning – an empirical study, IET Software, 3 (5), 2009, pp. 356–368.

[Gumm, 06] Gumm, D., Distribution dimensions in software development projects: A taxonomy, IEEE Software, 23 (5), 2006, pp. 45-51.

[Gupta, 2007] Gupta, A., & Seshasai, S., 24-Hour knowledge factory: using internet technology to leverage spatial and temporal separations, ACM Transactions on Internet Technology, 7 (3) 2007, 1-22.

[Gupta, 2009] Gupta, A., Deriving mutual benefits from offshore outsourcing: the 24-hour knowledge factory scenario, Communications of the ACM 52 (6), 2009, pp. 122–126.

[Hall, 02] Hall, T., Beecham, S., & Rainer, A., Requirements problems in twelve software companies: an empirical analysis, IEE Proceedings, 149 (5), 2002, pp. 153-160.

[Heeks, 01] Heeks, R., Krishna, S., Nicholsen, B., & Sahay, S., Synching or sinking: Global software outsourcing relationships, IEEE Software, 18 (2), 2001, pp. 54-60.

[Hefley, 06] Hefley, W.E., & Loesche, E.A., The eSCM-CL v1.1 Model Overview: The eSourcing Capability Model for Client Organizations v1.1. Carnegie Mellon University, IT Services Qualification Center, CMU-ITSQC-06-002. 2006.

[Herbsleb, 99] Herbsleb, J. D., & Grinter, R. E., Splitting the organization and integrating the code: Conway's law revisited, 21st International Conference on Software Engineering. IEEE Computer Society Press, Los Angeles, CA, 1999.

[Herbsleb, 01] Herbsleb, J. D., & Moitra, D., Global software development, IEEE Software, 18 (2), 2001, pp. 16-20.

[Herbsleb, 03] Herbsleb, J. D., & Mockus, A., An empirical study of speed and communication in globally-distributed software development, IEEE Transactions on Software Engineering, 29, 2003, pp. 1-14.

[Herbsleb, 07] Herbsleb, J. D., Global Software Engineering: The Future of Sociotechnical Coordination. In L. C. Briond and A. L. Wolf (Eds.) Proceedings of the International Conference on Software Engineering (pp. 188–198). 2007. Los Alamitos: IEEE Computer Society Press.

[Hernández-López, 10] Hernández-López, A., Colomo-Palacios, R., García-Crespo, A., & Soto-Acosta, P., Trust Building Process for Global Software Development Teams. A review from the Literature, International Journal of Knowledge Society Research, 1 (1), 2010, pp. 66-83.

[Highsmith, 01] Highsmith, J., & Cockburn, A., Agile software development: the business of innovation, Computer, 34 (9), 2001, pp. 120-127.

[Hinds, 03] Hinds, P. J., & Bailey, D. E., Out of sight, out of synch: Understanding conflict in distributed teams, Organization Science, 14 (6), 2003, pp. 615-632.

[Hsu, 10] Hsu, Y. L., Lee, C. H., & Kreng, V. B., The application of fuzzy Delphi method and fuzzy ahp in lubricant regenerative technology selection, Expert Systems with Applications, 37 (1), 2010, pp. 419-425.

[Hyder, 04] Hyder, E.B., Heston, K.M., & Paulk, M.C. (2004). The eSourcing Capability Model for Service Providers v2: Model Overview. Carnegie Mellon University, Institute for Software Research International, CMU-ISRI -04-113. 2004.

[Iden, 10] Iden, J., & Langeland, L., Setting the Stage for a Successful ITIL Adoption: A Delphi Study of IT Experts in the Norwegian Armed Forces, Information Systems Management, 27 (2), 2010, pp. 103-112.

[Jadhav, 09] Jadhav, A. S., & Sonar, R. M., Evaluating and selecting software packages: A review, Information and Software Technology, 51, 2009, pp. 555-563

[Jalote, 06] Jalote, P., & Jain, G., Assigning tasks in a 24-hour software development model, Journal of Systems and Software, 79, 2006, pp. 904-911.

[Jarvenpaa, 98] Jarvenpaa, S. L., & Leidner, D. E., Communication and trust in global virtual teams, Organization Science, 10 (6), 1999, pp. 791-815.

[Jiménez, 09] Jiménez, M., Piattini, M., & Vizcaíno, A., Challenges and Improvements in Distributed Software Development: A Systematic Review, Advances in Software Engineering, 2009, Article ID 710971.

[Karolak, 98] Karolak, D., Global Software Development: Managing Virtual Teams and Environments, IEEE Computer Society, Los Alamitos, CA, 1998.

[Karlsson, 07] Karlsson, L., Dahlstedt, A. G., Regnell, B., Natt och Dag, J., & Persson, A., Requirements engineering challenges in market-driven software development – An interview study with practitioners, Information and Software Technology, 49 (6), 2007, pp. 588-604.

[Kasi, 08] Kasi, V., Keil, M., Mathiassen, L., & Pedersen, K., The post mortem paradox: a Delphi study of IT specialist perceptions, European Journal of Information Systems, 17 (1), 2008, pp. 62-78.

[Kesner, 07], Kesner, R. M., & Castillo, R. R., The outsourcing of information technology services: a consideration of options and success factors, i-Manager's Journal of Management, 1, 2007, pp. 20-39.

[Kobitzsch, 01] Kobitzsch, W., Rombach, D., & Feldmann, R. L., Outsourcing in India, IEEE Software, 18 (2), 2001, pp. 78-86.

[Kommeren 07] Kommeren, R., & Parviainen, P., Philips experiences in global distributed software development, Empirical Software Engineering, 12 (6), 2007, pp. 647-660.

[Kotlarsky, 05] Kotlarsky, J., & Oshri, I., Social ties, knowledge sharing and successful collaboration in globally distributed system development projects, European Journal of Information Systems, 14 (1), 2005, pp. 37-48.

[Kotlarsky, 08] Kotlarsky, J., van Fenema, P. C., & Willcocks, L. P., Developing a knowledge-based perspective on coordination: The case of global software projects, Information & Management, 45 (2), 2008, pp. 96-108.

[Kraut, 95] Kraut, R. E., & Streeter, L., Coordination in software development, Communications of the ACM, 38 (3), 1995, pp. 69-81.

[Krishna, 04] Krishna, S., Sahay, S., & Walsham, G., Managing cross-cultural issues in global software outsourcing. Communications of the ACM, 47 (4), 2004, pp. 62-66.

[Landeta, 06] Landeta, J., Current validity of the Delphi method in social sciences, Technological Forecasting & Social Change, 73 (5), 2006, pp. 467-482.

[Layman, 06] Layman, L., Williams, L., Damian, D., & Bures, H., Essential communication practices for extreme programming in a global software development team, Information and Software Technology, 48 (9), 2006, pp. 781-794.

[Lehtola, 06] Lehtola, L., & Kauppinen, M., Suitability of Requirements PrioritizationMethods for Market-driven Software Product Development, Software Process Improvement and Practice, 11 (1), 2006, pp. 7-19.

[Liu, 10] Liu, S., Zhang, J., Keil, M., & Chen, T., Comparing senior executive and project manager perceptions of IT project risk: a Chinese Delphi study, Information Systems Journal, 20 (4), 2010, pp. 319-355.

[MacDuffie, 08] MacDuffie, J. P., HRM and distributed work, The Academy of Management Annals, 1 (1), 2008, pp. 549-615.

[Madachy, 08] Madachy, R. J., Cost Modeling of Distributed Team Processes for Global Development and Software-Intensive Systems of Systems, Software Process: Improvement and Practice, 13 (1), 2008, pp. 51-61.

[Marquardt, 01] Marquardt, M. J., Horvath, L., Global Teams: How Top Multinationals Span Boundaries and Cultures with High-Speed Teamwork, Davies-Black Publishing, Palo Alto, CA, 2001.

[Martins, 04] Martins, L. L., Gilson, L. L., & Maynard, M. T., Virtual Teams: What Do We Know and Where Do We Go From Here?, Journal of Management, 30 (6), 2004, pp. 805-835.

[Mattarelli, 09] Mattarelli, E., & Gupta, A., Offshore-onsite subgroup dynamics in globally distributed teams, Information Technology & People, 22 (3), 2009, pp. 242-269.

[Metiu, 06] Metiu, A., Owning the code: status closure in distributed groups, Organization Science, 17 (4), 2006, pp. 418-36.

[Milewski, 09] Milewski, A. E., Tremaine, M., Klober, F., Egan, R., Zhang, S., & O'Sullivan, P., Guidelines for Effective Bridging in Global Software Engineering, Software Process: Improvement and Practice, 13 (6), 2009, pp. 477-492.

[Momoh, 06] Momoh, J., & Ruhe, G., Release Planning Process Improvement – An Industrial Case Study, Software Process Improvement and Practice, 11 (3), 2006, pp. 295-307

[Mortensen, 01] Mortensen, M., & Hinds, P. J., Conflict and shared identity in geographically distributed teams, International Journal of Conflict Management, 12 (3), 2001, pp. 212-238.

[Muhairat, 10] Muhairat, M., Aldaajeh, S., Al-Qutaish, R.E., The Impact of Global Software Development Factors on Effort Estimation Methods, European Journal of Scientific Research, 46 (2), 2010, pp. 221-232.

[Myers, 97] Myers, M. D., Qualitative Research in Information Systems, MIS Quarterly, 21 (2), 1997, pp. 241-242.

[Nakatsu, 09] Nakatsu, R. T., & Iacovou, C. L., A comparative study of important risk factors involved in offshore and domestic outsourcing of software development projects: A two-panel Delphi study, Information & Management, 46 (1), 2009, pp. 57-68.

[Naranjo-Gil] Naranjo-Gil, D., Management information systems and strategic performances: The role of top team composition, International Journal of Information Management, 29 (2), 2009, pp. 104-110.

[Natt och Dag, 2005] Natt och Dag, J., Regnell, B., Gervasi, V., & Brinkkemper, S., A Linguistic-Engineering Approach to Large-Scale Requirements Management, IEEE Software, 22 (1), 2005, pp. 32-39.

[Naveh, 07] Naveh, Y., Richter, Y., Altshuler, Y., Gresh, D.L., & Connors, P., Workforce optimization: identification and assignment of professional workers using constraint programming, IBM Journal of Research and Development, 51(3), 2007, pp. 263-279.

[Ncube, 08] Ncube, C., Oberndorf, P., & Kark, A. W., Opportunistic Software Systems Development: Making Systems from What's Available, IEEE Software, 25 (6), 2008, pp. 38-41.

[Ngo-The, 09] Ngo-The, A., & Ruhe, G., Optimized Resource Allocation for Software Release Planning, IEEE Transactions On Software Engineering, 35 (1), 2009, pp. 109-123.

[Nicholson, 01] Nicholson, B., & Sahay, S., Some political and cultural issues in the globalisation of software development: case experience from Britain and India, Information and Organization, 11 (1), 2001, pp. 25-43.

[Okoli, 04] Okoli, C., & Pawlowski, S. D., The Delphi method as a research tool: an example, design considerations and applications, Information & Management, 42 (1), 2004, pp. 15-29.

[Olson, 00] Olson, G. M., & Olson, J. S., Distance matters, Human-Computer Interaction, 15 (2-3), 2000, pp. 139-178.

[Oshri, 07] Oshri, I., Kotlarsky, J., & Willcocks, L. P., Global Software Development: Exploring socialization in distributed strategic projects, Journal of Strategic Information Systems, 16 (1), 2007, pp. 25-49.

[Oshri, 08] Oshri, L., Kotlarsky, J., & Willcocks, L., Missing links: collaborative building social ties for global teamwork, Communications of the ACM, 51 (4), 2008, pp. 76–81.

[Oza, 06] Oza, N. V., Hall, T., Rainer, A., & Grey, S., Trust in software outsourcing relationships: An empirical investigation of Indian software companies, Information and Software Technology, 48 (5), 2006, pp. 345-354.

[Pauleen, 04] Pauleen, D. J., An inductively derived model of leader-initiated relationship building with virtual team members, Journal of Management Information Systems, 20 (3), 2004, pp. 227-256.

[Port, 09] Port, D., & Bui, T., Simulating mixed agile and plan-based requirements prioritization strategies: proof-of-concept and practical implications, European Journal of Information Systems, 18 (4), 2009, pp. 317–331.

[Prikladnicki, 03] Prikladnicki, R., Audy, J. L. N., & Evaristo, R., Global Software Development in Practice Lessons Learned, Software Process: Improvement and Practice, 8 (4), 2003, pp. 267-281.

[Prikladnicki, 09] Prikladnicki, R., & Audy, J. L. N., Comparing Offshore Outsourcing and the Internal Offshoring of Software Development: A Qualitative Study, AMCIS 2009 Proceedings, 2009, Paper 680.

[Ralité, 08] Ralité, J., Lamielle, X., Arni-bloch, N., & Léonard, M., Distributed information systems development: a framework for understanding and managing, International Journal of Computer Science and Applications, 5(3b), 2008, pp. 1-24.

[Ramasubbu, 05] Ramasubbu, N., Krishnan, M. S., & Kompalli, P., Leveraging global resources: A process maturity framework for managing distributed development, IEEE Software, 22 (3), 2005, pp. 80-86.

[Regnell, 01] Regnell, B., Höst, M., Natt och Dag, J., Beremark, P., & Hjelm, T., An Industrial Case Study on Distributed Prioritisation in Market-Driven Requirements Engineering for Packaged Software, Requirements Engineering, 6 (1), 2001, pp. 51-62.

[Robson, 02] Robson, C., Real World Research, Blackwell, Oxford, 2002.

[Rogers, 05] Rogers, P., & Lea, M., Social presence in distributed group environments: the role of social identity, Behaviour & Information Technology, 24, 2005, pp. 151-158.

[Ruhe, 05], Ruhe, G., & Saliu, M. O., The Art and Science of Software Release Planning, IEEE Software, 22 (6), 2005, pp. 47-53.

[Runeson, 06] Runeson, P., A Survey of Unit Testing Practices, IEEE Software, 23 (4), 2006, pp. 22-29.

[Sakthivel, 07] Sakthivel, S., Managing risks in offshore systems development, Communications of the ACM, 50 (4), 2007, pp. 69-75.

[Saliu, 05] Saliu, O., & Ruhe, G. Software release planning for evolving systems, Innovations in Systems and Software Engineering, 1 (2), 2005, pp. 189-204.

[Sawyer, 97] Sawyer, S., Farber, J., & Spillers, R., Supporting the social processes of software development teams, Information Technology & People, 10 (1), 1997, pp. 46-62.

[Sawyer, 00] Sawyer, S., Packaged software: implications of the differences from custom approaches to software development, European Journal of Information Systems, 9 (1), 2000, pp. 47-58.

[Schmidt, 01] Schmidt, R.C., Lyytinen, K., Keil, M., & Cule, P., Identifying software project risks: an international Delphi study, Journal of Management Information Systems, 17 (4), 2001, 5–36.

[Schümmer, 09] Schümmer, T., & Lukosch, S., Understanding Tools and Practices for Distributed Pair Programming, Journal of Universal Computer Science, 15 (16), 2009, pp. 3101-3125.

[Seshagiri, 06] Seshagiri, G., GSD: not a business necessity, but a march of folly, IEEE Software, 23 (5), 2006, pp. 63-64.

[Siakas, 06] Siakas, K.V., & Balstrup, B., Software outsourcing quality achieved by global virtual collaboration, Software Process: Improvement and Practice, 11(3), 2006, pp. 319–328.

[Sooraj, 08] Sooraj, P., & Mohapatra, P. K. J., Modeling the 24-h software development process, Strategic Outsourcing: An International Journal, 1 (2), 2008, pp. 122-141.

[Suzuki, 99] Suzuki, J., & Yamamoto, Y., Leveraging distributed software development, Computer, 32 (9), 1999, pp. 59-65.

[Svahnberg, 10] Svahnberg, M., Gorschek, T., Feldt, R., Torkar, R., Saleem, S. B., & Shafique, M. U., A systematic review on strategic release planning models, Information and Software Technology, 52 (3), 2010, pp. 237-248.

[Taxén, 06] Taxén, L., An integration centric approach for the coordination of distributed software development projects, Information and Software Technology, 48 (9), 2006, pp. 767-80.

[Treinen, 06] Treinen, J. J., & Miller-Frost, S. L., Following the sun: Case studies in global software development, IBM Systems Journal, 45 (4), 2006, pp. 773-783.

[Trienekens, 09] Trienekens, J. J. M., Kusters, R., Kriek, D., & Siemons, P., Entropy based software processes improvement, Software Quality Journal, 17 (3), 2009, pp. 231-243.

[Trigo, 10] Trigo, A., Varajão, J., Soto-Acosta, P., Barroso, J., Molina-Castillo, F. J., & Gonzalvez-Gallego, N., IT Professionals: An Iberian Snapshot. International Journal of Human Capital and Information Technology Professionals, 1 (1), 2010, pp. 61-75.

[van de Weerd, 06] van de Weerd, I., Brinkkemper, S., Nieuwenhuis, R., Versendaal, J., & Bijlsma, L., Towards a reference framework for software product management, Proceedings of the 14th International Requirements Engineering Conference, Minneapolis/St. Paul, Minnesota, USA, 2006, pp. 312-315.

[van de Weerd, 10] van de Weerd, I., Brinkkemper, S., & Versendaal, J., Incremental method evolution in global software product management: A retrospective case study, Information and Software Technology, 52 (7), 2010, pp. 720-732.

[van den Akker, 08] van den Akker, M., Brinkkemper, S., Diepen, G., & Versendaal, J., Software product release planning through optimization and what-if analysis, Information and Software Technology, 50 (1-2), 2008, pp. 101-111.

[Wallin, 02] Wallin, C., Ekdahl, F., & Larsson, S., Integrating Business and Software Development Models, IEEE software, 19 (6), 2002, pp. 28-33.

[Xu, 07] Xu, L., & Brinkkemper, S., Concepts of product software, European Journal of Information Systems, 16 (5), 2007, pp. 531-541.