SOFTWARE PROCESS IMPROVEMENT AND PRACTICE
Softw. Process Improve. Pract. 2009; **14**: 201–212
Published online 8 May 2009 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/spip.417

Benefits of Global Software Development: Exploring the Unexplored

Research Section

Eoin Ó Conchúir,[1]* Helena Holmström Olsson,[1,2]
Pär J. Ågerfalk[1,3] and Brian Fitzgerald[1]
[1] *Lero, The Irish Software Engineering Research Centre, University of Limerick, Ireland*
[2] *IT university, SE-412 96 Gothenburg, Sweden*
[3] *Uppsala University, Box 256, 751 05 Uppsala, Sweden*

Organizations are increasingly moving to the global software development (GSD) model because of significant benefits that can accrue. However, GSD is fraught with challenges arising from geographical, temporal and socio-cultural distances. The emphasis in the literature to date has typically been on how to overcome the challenges associated with GSD. While a number of GSD benefits have been widely referred to in the literature, there are also a number of less obvious benefits that can be inferred as potentially accruing from GSD. In this article, we identify the various benefits of GSD, labeling them as 'referred' and 'inferred', respectively. We provide a categorization in terms of (a) organizational, (b) team and (c) process/task. While the 'referred' benefits most often apply at the organizational level (e.g. cost savings, access to large multi-skilled workforces, reduced time to market and proximity to customer), the 'inferred' benefits apply to a greater extent at team and process/task level (e.g. task modularization, team autonomy, improved documentation and clearly defined processes). In the decision of whether or not to globalize software development activities, a categorization including both 'referred' and 'inferred' benefits will be helpful in providing a synthesis of all potential benefits associated with GSD. Copyright © 2009 John Wiley & Sons, Ltd.

KEY WORDS: global software development; benefits; challenges

1. INTRODUCTION

Global software development (GSD) is a phenomenon of increasing importance, given the perennial pressures to remain profitable and competitive in the global landscape. Companies can now leverage the emergence of large multi-skilled labor forces in lower-cost economies thanks to high-speed Internet-based communication links, through which the product, i.e. software code, can be quickly transferred between development sites. India and China, in particular, offer huge multi-skilled labor forces at greatly reduced cost compared with employment markets in the United States and western Europe. Other countries are also making an impact, such as Brazil, Eastern Europe and Russia, Malaysia and Vietnam.

GSD involves three types of distance: geographical, temporal and socio-cultural (Ågerfalk *et al.* 2005). Single teams can be separated by these distances, essentially becoming what is often termed

* Correspondence to: Eoin Ó Conchúir, Lero, The Irish Software Engineering Research Centre, University of Limerick, Ireland
† E-mail: eoin.oconchuir@lero.ie

'extended teams' or 'virtual teams'. In other circumstances, a single team may have all of its resources co-located, but with heavy reliance on other teams at remote locations. Vast geographical distances imply the difficulty of re-locating to another of the company's sites, and not being able to stroll over to a colleague's desk to chat about an implementation issue. Temporal distance across multiple time zones reduces the number of overlapping working hours, forcing a heavier reliance on asynchronous communication technologies. Socio-cultural distance arises from the different national and organizational backgrounds of the people involved and exacerbates communication breakdown.

Despite challenges related to these distances, major benefits have been associated with GSD. Apart from being a potential side-effect of mergers and acquisitions, GSD is posited as enabling major benefits such as lower development costs and access to huge multi-skilled labor forces, as already mentioned. However as researchers and practitioners have focused on overcoming GSD challenges, an exhaustive inventory of potential GSD benefits has not been compiled. While some benefits have been widely acknowledged in previous research, other potential benefits are evident but, nonetheless, overlooked to a large extent. In this article, we label these two categories of benefits as 'referred' and 'inferred', respectively. To elaborate slightly, we use the term 'referred' to cover the benefits which are quite widely referenced and known/accepted in the literature. We use the term, 'inferred', on the other hand, to cover those benefits which we have inferred or coalesced based on considering the challenges which need to be overcome for GSD and matching these with the types of competences and capabilities that can potentially emerge in these GSD contexts. For these inferred benefits, we report some research (Battin *et al*. 2001; Ebert and De Neve 2001; Espinosa and Carmel 2004; Carmel and Tija 2005; Delone *et al*. 2005; Herbsleb *et al*. 2005; Weakland 2005; Gumm 2006) which has helped stimulate our inference-making process. However, these inferred benefits have not been explicitly identified in the research referenced.

We then investigate what each benefit might offer companies aiming to leverage GSD. In doing so we present case study findings from two GSD companies, investigating to what extent the benefits identified are being realized in practice.

1.1. Challenges of GSD

Geographical, temporal and socio-cultural distances affect three important aspects of software development: communication, coordination and control (Ågerfalk *et al*. 2005). In fact, communication and control problems are recognized as being the most troublesome and pervasive in software development (Herbsleb *et al*. 1995).

A major challenge for GSD teams is the lack of informal communication which has been found to be essential in traditionally co-located development (Curtis *et al*. 1988; Carmel 1999). Written documentation is often inadequate when resolving misunderstandings, such as misunderstandings about requirements or changes in requirement specifications. Geographical and temporal distances make it more difficult to initiate contact with colleagues at other locations. While being indispensable for enabling face-to-face contact, the cost of travel can be prohibitive. A lack of overlapping working hours can lead to delays in feedback, rendering the development process less effective. Even a time zone difference of 1 hour can have a major effect when combined with different typical working hours in different countries. For example, a 1-hour difference in time zones can, due to different routines during a workday (e.g. different lunchtime norms), lead to only few overlapping hours and an appearance of higher than expected temporal distance (Herbsleb and Grinter 1999). Conversely, a European worker liaising with a counterpart in India who is working late shift may experience low temporal distance.

Socio-cultural distance can result in a fundamental difference in opinion about the nature of the software development process (Nicholson and Sahay 2001). It can lead to misunderstandings and non-native speakers struggling to follow technical discussions, especially over the phone. A general lack of familiarity with remotely located colleagues can result in a lack of 'teamness' and a reduced sense of trust. Also, while technical knowledge might be easy to obtain in a global labor pool, the difference in culture and experience might lead to difficulties in obtaining appropriate domain knowledge.

In summary, while software development is intrinsically a complex task, software development in a GSD context increases this complexity significantly, particularly with respect to communication, coordination and control issues.

Softw. Process Improve. Pract., 2009; **14**: 201–212
DOI: 10.1002/spip

1.2. Assumptions Made About GSD

Despite well-known challenges, GSD also presents practitioners with various benefits. As pointed out above, some of these are well known and quite widely cited, while some are not as obvious. Interestingly, the 'referred' benefits, which are generally considered to be the driving forces behind GSD, all seem to apply at the organizational level. That is, they contribute to top-level organizational goals, such as cost savings and increased efficiency. Admittedly, some of the 'inferred' benefits apply at the organizational level, but in addition we see these benefits as more directly affecting teams and basic software development processes and tasks. We would argue that the 'inferred' benefits should also be taken into consideration and that there is a need to highlight the full spectrum of GSD benefits.

Currently, there is a tendency to 'localize' GSD by attempting to reduce the geographical, temporal and socio-cultural distances involved. This approach assumes that the benefits of GSD do not fully justify truly GSD. Contrary to this, we have found teams shifting their working hours to increase the temporal overlap with remote colleagues, thereby aiming toward a 'virtual 8-hour day' (Holmström Olsson *et al*. 2008).

However, the decision of whether or not to globalize software development activities – or indeed the inclination to either 'localize' or fully leverage GSD – should be informed by the potential benefits it offers. We argue that this decision can be better informed if both 'referred' and 'inferred' benefits are taken into consideration. In the remainder of the article, we outline both the benefits, and provide a table in which we categorize all benefits, using the categories (a) organizational level, (b) team level and (c) process/task level.

2. THE 'REFERRED' BENEFITS OF GSD

In this section, we briefly outline well-known benefits of GSD. These tend to apply at an organizational level and have been previously acknowledged in research.

2.1. Cost Savings

Perhaps the original and most sought-after benefit of GSD has been that of reduced cost of development (Carmel and Agarwal 2001). The basis for this benefit is that companies are globalizing their software development activities to leverage cheaper employees located in lower-cost economies. This has been made possible by the deployment of cross-continental high-speed communication links enabling the instantaneous transfer of the basic product at hand: software.

The difference in wages across regions can be significant, with a US software engineer's salary being multiple times greater than that of a person with equivalent skills in (at least parts) of Asia or South America. In 2005, the annual base pay of a software engineer located in India was $10,300 (Mercer: China and India: Comparative HR Advantages 2005). However, this seems to be rising and there has been hypergrowth in local IT employment markets such as in Bangalore. It is our experience that companies are now looking at alternative locations, which offer more acceptable attrition rates with the continued promise of cheaper labor.

2.2. Access to Large Multi-skilled Workforces

GSD provides the unprecedented possibility to leverage large pools of skilled labor by coordinating across distance (Grinter *et al*. 1999; Ebert and De Neve 2001; Herbsleb and Moitra 2001; Damian *et al*. 2003). Companies have the opportunity to expand their software development activities to include the contributions of thousands of technically skilled workers, wherever they may be located, to form virtual global corporations (Suzuki and Yamamoto 1999; Herbsleb and Moitra 2001; Carmel and Tija 2005).

2.3. Reduced Time to Market

A controversial benefit of GSD has been that of the 'follow-the-sun' approach, described in detail by Carmel (1999). Time zone effectiveness is the degree to which an organization manages resources in multiple time zones, maximizing productivity by increasing the number of hours during a 24-hour day that software is being developed by its teams. When time zone effectiveness is maximized to span 24 hours of the day, this is referred to as the 'follow-the-sun' development model. This is achieved by handing off work from one team at the end of their day to another team located in another time zone. The approach can aid organizations which are

DOI: 10.1002/spip

under severe pressure to improve time to market (Herbsleb and Moitra 2001).

2.4. Proximity to Market and Customer

By establishing subsidiaries in countries where the company's customers are located, GSD allows it to develop software close to their customers and to increase knowledge of the local market (Herbsleb and Moitra 2001). Creating new jobs can create good will with local customers, possibly resulting in more contracts (Ebert *et al*. 2001). Indeed, it may be a business necessity to locate closer to customers in order to expand to other markets. For example, a company that develops software for embedded systems may focus on large manufacturing companies based in China or a software automotive company may locate part of the development in Germany. Development activities may even be located on the same campus as the organization's large customer. Companies may also look to establishing strategic partnerships to gain access to new markets (Karolak 1998).

3. THE 'INFERRED' BENEFITS OF GSD

Above, we have highlighted four well-known benefits that have been cited as driving forces toward the globalization of software development activities. However, there have been individual reports of additional benefits that may be realized through GSD. Up until now, these benefits have been mostly overlooked. Indeed, and as the label reflects, the benefits covered below are not as obvious as the 'referred' benefits mentioned earlier. While the 'referred' benefits tend to easily affect the overall company policy, we believe that additional and sometimes more indirect benefits may offer great potential and indeed contribute in strategic company decisions.

While some of the 'inferred' benefits we identify are applicable at organization level (as is the case with 'referred' benefits), they also seem to affect lower level coordination and collaboration within and between GSD software teams as well as the basic software engineering processes and tasks at hand. For the purpose of this article, we therefore use the categories organization, team and process/task when discussing the identified benefits. By focusing on leveraging the full range of benefits, we argue that companies may reap more rewards from their GSD activities, and that GSD may not need to be seen as only a challenge – but also as a potential opportunity – for software development companies.

3.1. Organizational Benefits

Two of the 'inferred' benefits apply primarily at the organizational level. We refer to these as 'innovation and shared best practice' and 'improved resource allocation'.

3.1.1. Innovation and Shared Best Practices

The global business environment demands and expects innovative, high-quality software that meets its needs (Highsmith and Cockburn 2001; Scacchi *et al*. 2006). Organizations can take advantage of increased innovation and shared best practice that arises from the collaboration of team members who come from different national and organizational backgrounds (Ebert and De Neve 2001; Carmel and Tija 2005).

In large complex organizations, decentralized, independent individuals interact in self-organizing ways to create innovative and emergent results (Highsmith and Cockburn 2001). Such organizations base their success on innovation and their innovation capabilities come from talent – from their most brilliant, intelligent and creative engineers. Companies that expand into other countries in order to tap into talent have been termed 'knowledge seekers' (Chung and Alcacer 2003). Such organizations tend to act somewhat differently compared to organizations that offshore purely for cost reasons (Carmel and Tija 2005) and we can see an acknowledgement of this benefit through the action of such companies.

3.1.2. Improved Resource Allocation

As an extension to the benefit of access to large multi-skilled labor pools, the organization can benefit from the influx of new (lower cost) labor in other countries. As a result of this, organizations can benefit from re-assigning the newly redundant higher cost resources to other, often more strategic, activities while also avoiding the employee turmoil and backlash associated with workforce reductions (Weakland 2005). Changes in allocation can adhere to the challenge of replacing isolated expertise and

DOI: 10.1002/spip

instead create skill-broadening tasks and effective teamwork (Ebert and De Neve 2001).

3.2. Team Benefits

At the team level, we find three inferred benefits, namely 'improved task modularization', 'reduced coordination cost', and 'increased team autonomy'.

3.2.1. Improved Task Modularization

According to Conway's Law, the structure of the system mirrors the structure of the organization that designed it (Herbsleb and Grinter 1999). In fact, it is the product architecture that should determine the team structure, rather than the other way around (Carmel 1999). In earlier work we have seen the importance of a separation of concerns when decomposing work into modules (Parnas 1972), and it appears that these principles could be extremely relevant for managing coordination complexity.

The nature of GSD assumes teams to splitting their work across feature content into well-defined independent modules (Ebert and De Neve 2001; Sahay 2003; Bass and Paulish 2004), without 'stepping on each other's toes' (Carmel 1999). This allows decisions to be made about each component in isolation (Herbsleb and Grinter 1999). Partitioning work tasks horizontally results in each site having responsibility for the whole lifecycle of particular functions/modules, it decreases interdependencies and hence, coordination costs (Battin *et al*. 2001). For example, source code branching enables software development teams to work on source code in parallel, and merging the sections once they have been developed (Herbsleb *et al*. 2005).

3.2.2. Reduced Coordination Cost

While we acknowledge that temporal distance can prove to be a challenge for GSD teams, it can also be seen as beneficial in terms of coordination. Coordination costs are reduced when team members are not working at the same time (Espinosa and Carmel 2004). The producer of a unit of work can complete the work during the off-hours of the person who requested that work. In essence, coordination costs are reduced since no direct coordination takes place when two people are not working at the same time. However, this benefit will only be realized if requirements are clear and if the two parties have agreed on the unit of work that will be developed.

3.2.3. Increased Team Autonomy

Gumm (2006) found that organizational and geographical distribution of 'software development units' imply a certain degree of autonomy for each unit. The study reported that this autonomy allowed for the necessity to maintain the different working cultures of each team, e.g. team jargon, working routines and processes etc. This was viewed as necessary to maintain the quality of the work of a single team even if this in turn required careful synchronization of the single processes.

3.3. Process/Task Benefits

In addition to the organizational and team-oriented inferred benefits outlined earlier, there are three further inferred benefits that apply primarily at the process/task level. We refer to these as 'formal record of communication', 'improved documentation', and 'clearly defined processes'.

3.3.1. Formal Record of Communication

As asynchronous communication relies on technologies such as e-mail and fax (Kiel 2003; Boland and Fitzgerald 2004), a written communication history is usually left, showing who said what, who were involved in a discussion and at what time the discussion was held. (Carmel and Agarwal 2001; Damian and Zowghi 2002). This record allows for increased traceability and accountability (Ågerfalk 2004). Also, asynchronous communication allows for input from diverse stakeholders irrespective of geographical location (Damian and Zowghi 2002). In a study by Gumm (2006), the findings show that asynchronous communication technologies were beneficial since they allow for people to think about a problem before they ask. Respondents also emphasized that asynchronous communication gave them better opportunities for thinking intensively about a problem before answering a question from a remote colleague.

3.3.2. Improved Documentation

Delone *et al*. (2005) state that distributed teams have an increased focus on documentation in order to aid their communication. Gumm (2006) reported this as an advantage, in that documentation is better supported within distributed project settings. Information is documented and distributed electronically rather than discussed face-to-face, which

DOI: 10.1002/spip

allows for the passing on of project-specific knowledge in distributed settings.

3.3.3. Clearly Defined Processes

Independent of a project's process maturity, the definition and structuring of processes is a challenge (Gumm 2006). While distributed project settings seem to challenge process maturity, they also seem to support it. Process definitions are compiled more carefully in distributed settings. It was noted that if team members were co-located, much of the processes would probably not be formalized. In a GSD setting, however, processes need to be formally documented and described in such a manner that many different teams can understand them. While this can be seen as a precondition for GSD, it is also an effect in that the GSD environment encourages organizations to carefully define their processes already in an initial stage.

4. RESEARCH METHOD

After an extensive literature study in which 'referred' and 'inferred' benefits were identified, the goal of our research was to understand the extent to which they are being realized. To understand this, a case study approach was taken (Yin 2003) and qualitative interviews were conducted at Pennysoft (a US-based financial company; the original names of the companies have been anonymized) and at Semicon (a US-based manufacturer of chips and computer, networking and communications products). Case studies can be very valuable in generating an understanding of the reality of a particular situation, and can provide a good basis for discussion. The approach has been widely used in research seeking to understand phenomena through the meanings that people assign to them (Orlikowski and Baroudi 1991; Adam and Fitzgerald 2000) and the process whereby computer information systems influence and are influenced by a specific context.

4.1. Research Sites

Pennysoft provides financial services and investment resources. The company was chosen due to its large-scale software development activities that are spread across the United States, Ireland and India. While software development is not on its core business, software systems play a large role in providing a capable infrastructure for the provision of their services. Its software development activities span multiple locations in the United States, Ireland and India, making it a true practitioner of GSD. The main focus of the study was a Pennysoft site in Ireland. Ireland's role is central in the globally dispersed teams, as it deals directly with the United States and India. In many cases, the requirements are generated in the United States, with software development then taking place in both the United States and Ireland. Most quality assurance (QA) activities take place at the Indian sites. The Indian sites became involved in June 2005.

Semicon, a NASDAQ-quoted company, is a leading manufacturer of chips and computer, networking and communications products. The focus of our study was one of Semicon's Irish sites which employs 125 people. The software being developed here 'facilitates' the silicon products being produced by the company, allowing third-party vendors to access the functionality of the hardware. The software program manager resides at the case study site, and manages multiple sites including India, Poland, China and Malaysia. Software requirements are provided by the marketing group. Development work is then planned and much effort is spent in designing the feature-based components of the system. Each development team is then given a feature set to work on. The teams work relatively independently because of the difficulty of communicating and coordinating across global distances. Upon completion of the project, key managers and architects travel to the case study site in order to integrate the work that was completed by each remote team.

4.2. Research Design

Given that relatively little research has been conducted into the benefits of GSD, the aim of this study was to provide a better understanding of those benefits. As such, the study was exploratory and qualitative in nature. A case study approach was adopted because of its emphasis on context, with an ability to develop a better understanding through the collection of 'thick' description of the phenomenon of interest (Yin 2003). The benefits

Softw. Process Improve. Pract., 2009; **14**: 201–212
DOI: 10.1002/spip

identified in the literature were used as a basis for the interview protocol, as well as for analyzing the empirical findings. Such an approach is recommended by Patton (1990), who argues that an interview guide is useful for focusing interviews and can also be used as a descriptive framework for analysis.

Data was collected over a period of 3 years and involved semi-structured interviews as well as workshops and on-site meetings at the two company sites. Table 1 provides a summary of the research activities involved.

Given the exploratory nature of the study, the interviews included open-ended questions, where interviewees were given a chance to explore their thoughts on their GSD experiences. Interviewees were selected to provide a mix of experience in different projects at various levels of responsibility, i.e. project managers, software engineers, product managers etc. The selection of interviewees was opportunistic, in that employees visiting the Irish site from other sites were interviewed where appropriate. A total of 13 people were interviewed in Pennysoft, 3 of whom are based in India and 1 US-based interviewee. A total of 10 people were interviewed in Semicon. Several interviewees were interviewed twice due to the changing nature of their roles during the study. All our interviews were analyzed using the three categories that emerged from the literature study, i.e. organizational, team and process/task. By mapping the interview findings to these overall categories, we can provide a comprehensive overview including both 'referred' and 'inferred' benefits which will provide useful insights for companies interested in adopting the GSD development model.

It is worth mentioning that while case-based research adds depth and provides a rich understanding of the phenomena under study, it is always questionable to what extent the results can be generalized to other contexts (Lee and Baskerville 2003; Yin 2003). Rigorous case-based research could be viewed as explorative research that generates 'well-founded but as-yet untested hypotheses' (Lee and Baskerville 2003). Future research could use our results as a basis for large-scale investigations aiming to explore the transferability of our findings.

Table 1. Summary of research activities

Company	Date	Research activity
Pennysoft	January 2005	Workshop on GSD
	March 2005	On-site meeting with company management
	July 2005	Interviews with three project managers, technical product manager
	April 2006	Interviews with principal engineer, project leader (India-based), project manager. Telephone interviews with senior systems analyst, director of software management, senior software engineer, software developer (India-based), project manager
	July 2006	Workshop at university exploring findings (also attended by Semicon)
	April 2008	Interviews with two project managers (one of whom is US-based), software engineer (India-based)
Semicon	July 2005	On-site meeting with company management.
	August 2005	Interviews with software engineer, software project manager, general manager, member of technical staff
	April 2006	Workshop on offshoring/outsourcing at university
	June 2006	Interviews with product manager, technical leader, team leader, engineering project leader, project manager, staffing manager
	July 2006	Workshop at university exploring findings (also attended by Pennysoft)
	August 2006	On-site workshop with company management feeding back results

5. RESULTS

This section presents the findings of the case study at Pennysoft and Semicon, according to each of the 'referred' and 'inferred' benefits as identified above.

5.1. The 'Referred' Benefits of GSD

5.1.1. Cost Savings

Every Pennysoft interviewee shared the opinion that Pennysoft was distributing their software development to ultimately reduce costs. These costs include cost-effective labor, but also cheaper real estate and infrastructure. There is a general policy in the company to 'offshore' work. Indeed, the Irish site was founded primarily as a cost-saving measure by the company (in the meantime, the site's

DOI: 10.1002/spip

cost has risen). According to several interviewees, teams have been mandated to offshore 40% of their software development activity. An insight into the company's cost savings is the per hour rate billed by each site. Ireland charges 72% of the US rate, while India charges just 32% of the US rate. For Semicon, the project costs are broadly similar, with the Irish site costing 75% compared to the US cost and Indian projects costing 25% of the US cost.

5.1.2. Access to Large Multi-skilled Workforces

When asked whether access to a large skilled offshore labor pool was of benefit to the company, Pennysoft interviewees generally agreed that such a benefit was more of a facilitator of reduced development costs. As such, Pennysoft is not being forced to seek skilled labor in new employment markets due to a skill shortage in the United States. It was recognized by interviewees that the benefit of access to a large skilled labor pool is realized incrementally, in that it takes several years for a site to become mature and efficient in software development. An interviewee noted that this benefit may now be more realistic than several years ago, as local employment markets such as Bangalore had been suffering from high rates of employee attrition.

This was an important factor for Semicon when seeking to set up new development sites. While seeking relatively cheaper labor, they must also take into account the need for employing highly skilled employees. The program manager interviewed indicated that the company can access 'genius employees' in countries such as Malaysia, China and India.

5.1.3. Reduced Time to Market

This proposed benefit was rejected by the Pennysoft interviewees. The team structure at Pennysoft involves 'extended teams', with teams consisting of members from the United States, Ireland and India. Interviewees accepted that a model such as 'follow-the-sun' may be appropriate for a project at maintenance phase, as work tasks are relatively small and can be handed to one person at a time. However, the model does not suit complex software development tasks. The teams suffer from being distributed across multiple time zones. Instead of leveraging that temporal distance, the team members shift their working hours in order to maximize their temporal overlap with remote colleagues.

Unlike the Pennysoft teams, the teams at Semicon are designed to be self-sufficient units of co-located colleagues. As such, any one team will only have team members at one single location. As previously described, Semicon teams work independently on feature sets and ultimately supply the program with completed components. This has an advantage of reducing time to market insofar as each component can function relatively independently of the release schedule of other teams.

5.1.4. Proximity to Market and Customer

Rather than the benefit of being able to locate software development activity closer to customers, Pennysoft has in fact used GSD to stretch their software development activities away from their traditional US-based customers. Initially, development work was moved away from the company's home city. When the Irish sites, and later the Indian sites, were established by Pennysoft, the customer-facing team members remained at the original site, maintaining essential communication with its customers.

In contrast to this, Semicon does use GSD to locate their development activities closer to their customers. As a hardware manufacturer, it is important for the company to have access to Asian manufacturing-based markets. Semicon's Chinese development teams, in particular, allow for the company to reduce visible socio-cultural distance between them and their potential customers.

5.2. The 'Inferred' Benefits of GSD

5.2.1. Innovation and Shared Best Practices

Pennysoft interviewees did not mention this benefit without being prompted to do so. There was a mixed response to their opinion on the benefit. For example, a project manager felt that having a mixture of people from different backgrounds '*will always help... different views by different people*'. However, another project manager was dismissive of the usefulness of the best practices being sent from an Indian site.

No Semicon interviewee agreed that innovation spreads between teams due to the mixture of nationalities. This could perhaps be down to the fact that Semicon teams are in single locations, and that daily contact between teams are kept to a minimum. The 'localized' development at Semicon may be limiting the globalized benefit of innovation arising from different national backgrounds.

Softw. Process Improve. Pract., 2009; **14**: 201–212
DOI: 10.1002/spip

5.2.2. Improved Resource Allocation

Pennysoft's approach to team structure allows for human resources to be included in one team wherever they may be located. This is of benefit for Pennysoft, as teams have been able to incorporate lower-cost engineers into their teams without being limited to their geographic location. Resources can also be moved between projects within the same business group as they become available, maximizing the company's usage of its human resources.

Semicon cannot benefit from this same approach, as all team members must be co-located. However, resource allocation is flexible in that the team structure allows for entire third-party teams to be involved in product development. This allows Semicon to leverage outsourcing providers without greatly affecting the existing team structure.

5.2.3. Improved Task Modularization

Pennysoft does not generally realize the benefit of modularization of software development according to functional parts of the system, although it was recognized that such an approach would be advantageous to reduce the negative effect of working across distance. However, different skill sets are found at different sites and work tasks are distributed accordingly. Also, the globally distributed teams are relatively small. Interviewees felt that the smaller team size and the nature of their distributed software development work did not allow for tasks to be divided into subsets of responsibility.

In contrast to Pennysoft, Semicon does realize this benefit. Semicon teams are perhaps more mature and established, which allows for them to work relatively independently of each other. Task modularization reduces the need for cross-team communication, and so reduces the coordination complexity involved with development activities. However, it should be noted that Semicon invests a large amount of time into the planning and design phases in order to allow for their teams to work on well-defined feature sets with explicit interfaces with other components.

5.2.4. Reduced Coordination Cost

A Pennysoft interviewee based in India agreed with this benefit, as meetings with other sites would be limited to several hours of their day, whereas the '*real*' work could be completed at other times. However, the Irish project managers did not realize this benefit. They expressed dissatisfaction with having to coordinate with India during their morning, and then with the United States during their afternoon. The continuous temporal overlap with other sites made this benefit impossible to realize.

This benefit was realized by Semicon, which relates to *improved task modularization* above. Previously, Semicon development projects spanned as many as eight sites across three continents. Management found this to be highly complex and not worthwhile. To counter this coordination complexity, all projects were limited to two 'geographies' (or locales) only. Along with co-located team members, this meant that coordination cost was reduced, hence improving the efficiency of the GSD projects.

5.2.5. Increased Team Autonomy

Pennysoft interviewees rejected the autonomy implied by the distance between sites. Rather, it seemed that the Indian sites currently lack the experience which would allow them to become more autonomous. This may change in years to come as the sites become more mature, therefore requiring less day-to-day management from the Irish site, as happened with the United States and Ireland during the previous decade.

Semicon teams are given the mandate to work independently of other teams, therefore implying a high degree of autonomy for each team. The benefits of such a structure have already been discussed.

Finally, the overall sentiment of the interviewees was similar for the following three proposed benefits: *formal record of communication; improved documentation;* and *clearly defined processes*. Interviewees felt that clearly defined processes are a prerequisite for success in GSD. The distributed context also leads to the formalization of communication and to increased focus on documentation. However, interviewees generally questioned such proposed benefits, as the global distribution demands more effort than a co-located environment. For example, shared documentation implies additional coordination dependency between sites, and the maintenance of existing documentation is made harder by the distributed nature of the stakeholders. For Semicon specifically, clearly defined processes were not so important at a team level. This is due to the relative independence of each team, allowing them

DOI: 10.1002/spip

to adopt team-specific tools and processes as necessary. While clearly defined cross-site processes were important for Pennysoft, the Irish and Indian teams found it difficult for their suggested processes to be adopted by the already well-established US-based teams.

6. CONCLUSIONS

As recognized in this article, some benefits of GSD have been widely cited and can be considered known to both researchers and practitioners. In this study, we outline these as 'referred' benefits of GSD. However, additional benefits are evident and they have been, to some extent, overlooked. In this article, we have identified these 'inferred' benefits in order to provide a synthesis of GSD benefits. This will hopefully lead to a more informed debate on the topic as well as more informed decisions on whether or not to pursue GSD. As can be seen, a majority of the 'referred' benefits apply at the organizational level while the 'inferred' benefits that we have identified apply at a team or at a process/task level. This is probably part of the reason for them not being widely acknowledged as driving factors toward GSD. See Table 2 for a summary of our synthesis of the benefits offered by GSD, structured according to the categories of (a) organizational level, (b) team level and (c) process/task level. We also report the extent of their realization in our case study companies.

Table 2. GSD benefits and extent of realization

Benefits	Pennysoft	Semicon
Organizational level		
Cost savings	Y	Y
Access to large multi-skilled workforces	P	Y
Reduced time to market	N	Y
Proximity to market and customer	N	Y
Innovation and shared best practice	P	N
Resource allocation	Y	N
Team level		
Improved task modularization	N	Y
Reduced coordination cost	P	Y
Increased team autonomy	N	Y
Process/task level		
Formal record of communication	P	P
Improved documentation	P	P
Clearly defined processes	P	P

Y, benefit realized in company; P, benefit partially realized and N, benefit not realized.

We have also pointed out the on-going struggle between reducing the distances of GSD and making the most of the dynamic context of the global environment. For example, we see attempts to reduce coordination costs by effective modularizing work and shifting working hours, while at the same time wishing to leverage GSD by sharing innovation and best practice between teams. The debate on this matter up until now has not been informed by a full synthesis of the benefits. Often, benefits at the organizational level have been acknowledged as driving forces, while benefits at a team or process/task level have been overlooked. We believe that in acknowledging also these 'inferred' benefits we can stimulate a more informed debate on this matter.

Cost-benefit trade-offs in GSD are still not fully understood (Espinosa and Carmel 2004). The GSD community has yet to come to a consensus on which benefits are realistic, and whether or not practitioners should aim for the realization of each of them. For example, it is not yet clear to what extent cost savings can and are being realized. While salary savings are obvious in most cases, our interviewees emphasize that estimations of coordination costs are far more complex and that misunderstandings that might affect time to market are difficult to count for. While this situation might be evident in co-located teams, it is especially true in extended teams, i.e. teams spanning several geographical locations. Also, the follow-the-sun concept has been dismissed by many, but is still being promoted (see e.g. Carmel 2007). Most probably, certain benefits may only be realistic in specific contexts while some benefits may be mutually exclusive.

REFERENCES

Adam F, Fitzgerald B. 2000. The status of the IS field: historical perspective and practical orientation. *Information Research* **5**(4): http://informationr.net/ir/5–4/paper81.html/.

Ågerfalk PJ. 2004. Investigating actability dimensions: a language/action perspective on criteria for information systems evaluation. *Interacting with Computers* **16**(5): 957–988.

Ågerfalk PJ, Fitzgerald B, Holmström H, Lings B, Lundell B, Ó Conchúir E. 2005. A framework for considering opportunities and threats in distributed software development. *International Workshop on*

DOI: 10.1002/spip

Distributed Software Development. Austrian Computer Society: Paris.

Bass M, Paulish D. 2004. Global software development process research at siemens. In *International Workshop on Global Software Development*, Edinburgh, Scotland, 24th May.

Battin RD, Crocker R, Kreidler J, Subramanian K. 2001. Leveraging resources in global software development. *IEEE Software* **18**(2): 70–77.

Boland D, Fitzgerald B. 2004. Transitioning from a Co-located to a globally-distributed software development team: A case study and analog devices Inc. *3rd International Workshop on Global Software Development*, Edinburgh, Scotland.

Carmel E. 1999. *Global Software Teams: Collaborating Across Borders and Time Zones*, 1st edn. Prentice Hall: Upper Saddle River, NJ.

Carmel E. 2007. Keynote speech. In *International Conference on Global Software Engineering (ICGSE)*, Munich, Germany, August 27–30.

Carmel E, Agarwal R. 2001. Tactical approaches for alleviating distance in global software development. *IEEE Software* **18**(2): 22–29.

Carmel E, Tija P. 2005. *Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce*. Cambridge University Press: Cambridge.

Chung W, Alcacer J. 2003. Knowledge sources and foreign investment location in the US. In *Conference of the Academy of International Business*, Monterrey, California.

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

Damian D, Lanubile F, Oppenheimer HL. 2003. Addressing the challenges of software industry globalization: The workshop on global software development. *25th International Conference on Software Engineering*. IEEE Computer Society: Portland.

Damian D, Zowghi D. 2002. The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization. *IEEE Joint International Conference on Requirements Engineering*. IEEE Computer Society: Los Alamitos.

Delone W, Espinosa JA, Lee G, Carmel E. 2005. Bridging global boundaries for IS project success. *Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) – Track 1*, Vol. 01. Los Alamitos, California. IEEE Computer Society.

Ebert C, De Neve P. 2001. Surviving global software development. *IEEE Software* **18**(2): 62–69.

Ebert C, Parro CH, Suttels R, Kolarczyk H. 2001. Improving validation activities in a global software development. In *Proceedings of the 23rd International Conference on Software Engineering*, Toronto.

Espinosa JA, Carmel E. 2004. The effect of time separation on coordination costs in global software teams: A dyad model. *37th Hawaiian International Conference on System Sciences*. IEEE: Big Island, Hawaii.

Grinter RE, Herbsleb JD, Perry DE. 1999. The geography of coordination: Dealing with distance in R&D work. In *International Conference on Supporting Group Work 1999*, 306–315.

Gumm D. 2006. Distribution dimensions in software development projects: A taxonomy. *IEEE Software* **23**(5): 45–51.

Herbsleb JD, Grinter RE. 1999. Splitting the organization and integrating the code: Conway's law revisited. *21st International Conference on Software Engineering*. IEEE Computer Society Press: Los Angeles, CA.

Herbsleb JD, Klein H, Olson GM, Brunner H, Olson JS, Harding J. 1995. Object-oriented analysis and design in software project teams. *Human-Computer Interaction* **10**: 249.

Herbsleb JD, Moitra D. 2001. Guest Editors' introduction: Global software development. *IEEE Software* **18**(2): 16–20.

Herbsleb JD, Paulish DJ, Bass M. 2005. Global software development at siemens: Experience from nine projects. *27th International Conference on Software Engineering*. ACM: St. Louis, MO.

Highsmith J, Cockburn A. 2001. Agile software development: the business of innovation. *Computer* **34**(9): 120–127.

Holmström Olsson H, Ó Conchúir E, Ågerfalk PJ, Fitzgerald B. 2008. Two-stage offshoring: An investigation of the Irish Bridge. *MIS Quarterly* **32**(2): pp 1–23.

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

Kiel L. 2003. Experiences in distributed development: A case study. In *ICSE International Workshop on Global Software Development*, Portland, Oregon.

Lee AS, Baskerville RL. 2003. Generalizing generalizability in information systems research. *Information Systems Research* **14**(3): 221–243.

DOI: 10.1002/spip

Mercer: China and India: Comparative HR Advantages. 2005. http://www.mercer.com/china-indiareport, accessed 27th February 2006.

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

Orlikowski WJ, Baroudi JJ. 1991. Studying information technology in organizations: Research approaches and assumptions. *Information Systems Research* **2**(1): 1–28.

Parnas DL. 1972. On the criteria to be used in decomposing systems into modules. *Communications of the ACM* **15**(12): 1053–1058.

Patton MQ. 1990. *Qualitative Evaluation and Research Methods*, 2nd edn. Sage Publications: Thousand Oaks, CA.

Sahay S. 2003. Global software alliances: the challenge of 'standardization'. *Scandinavian Journal of Information Systems* **15**: 3–21.

Scacchi W, Feller J, Fitzgerald B, Lakhani K, Hissam S. 2006. Understanding free and open source software processes, *Software Process: Improvement and Practice*, Vol. 11.

Suzuki J, Yamamoto Y. 1999. Leveraging distributed software development. *Computer* **32**(9): 59–65.

Weakland T. 2005. *Global IT Outsourcing Study*. DiamondCluster International, Inc.: http://www.diamondconsultants.com/PublicSite/ideas/perspectives/downloads/Diamond2005OutsourcingStudy.pdf, accessed 15th January 2009 (2005).

Yin RK. 2003. *Case Study Research: Design and Methods*. Sage Publications inc: Thousand Oaks, CA.

DOI: 10.1002/spip