9 TEN STRATEGIES FOR SUCCESSFUL DISTRIBUTED DEVELOPMENT

Brian Lings
Björn Lundell
University of Skövde
Skövde, Sweden

Pär J. Ågerfalk
Brian Fitzgerald
University of Limerick
Limerick, Ireland

Abstract   *This paper presents an overview of the field of distributed development of software systems and applications (DD). Based on an analysis of the published literature, including its use in different industrial contexts, we provide a preliminary analysis that structures existing DD knowledge, indicating opportunities but identifying threats to communication, coordination, and control caused by temporal distance, geographical distance, and socio-cultural distance. An analysis of the case and field study literature has been used to identify strategies considered effective for countering the identified threats. The paper synthesizes from these a set of 10 general strategies for successful DD which, if adopted, should lead to increased company resilience.*

Keywords   Distributed software development, global software development, strategies, case studies, distributed development framework, development process, literature analysis

1   INTRODUCTION

Resilience—the ability to adapt to changing circumstances and recover from disruption—is an important property of organizations in today's turbulent business environment (Lengnick-Hall 2005; Riolli and Savicki 2003). In the wake of the bursting information technology bubble, many software organizations have turned toward glo-

bally distributed development (DD) as a way of cutting costs, gaining access to new markets, and enabling round-the-clock work (e.g., Carmel 2003). This is a trend that is likely to continue: according to the United Nations' 2004 World Investment Report, off-shoring of IT-enabled services is forecast to expand 24-fold by 2007 from a base of $1 billion in 2002. However, DD is in itself a rather disruptive innovation (Lyytinen and Rose 2003), putting new demands on both individuals and organizations. In any case, DD is certainly not the "silver bullet that slays the software productivity monster," alluding to Fred Brooks' vivid description of the software crisis (Brooks 1986, p. 1071). On the contrary, there are many issues to tackle for any organization adopting DD.

In ideal software development teams, members have rich interactions, both formal and informal; share a common organizational culture, which promotes good coordination and facilitates effective control; represent a good mix of all required technical skills and relevant experience, made readily accessible to all team members; and are familiar with, and provided with, homogeneous tools and technologies appropriate for the project. DD adds new demands to the software development process by potentially threatening each of these ideal properties.

In this paper we characterize the main opportunities and threats to DD projects, and synthesize, from reported case and field studies in real industrial settings, strategies which have proven successful in practice. These form 10 general strategies for successful DD.

The paper is organized as follows. Section 2 presents the concepts used in the analysis, and outlines a framework used for characterizing opportunities and threats in DD. Section 3 presents the research approach adopted for this study. Section 4 presents 10 major strategies that together represent a synthesis of those proposed in the literature based on case and field studies. In section 5 we summarize and reflect on our findings.

2   THEORETICAL BACKGROUND

For the purpose of this research, we take the position of Ågerfalk et al. (2005) in defining DD. Here, *development* is interpreted broadly as any software development life cycle activity. This thus extends beyond "pure" development activities and includes, for example, deployment and maintenance. A development team is distributed if its team members are not collocated, but geographically spread out.

For a number of years, the international workshop on Global Software Development (GSD) has highlighted the impact of distribution on *communication*, *coordination*, and *control* within DD life cycle activities (see, for example, Damian et al. 2003). This view is consistent with the position taken by a number of authors who have focused on one or more of these three fundamental processes to understand DD (e.g., Carmel and Agarwal 2001; Evaristo et al. 2004; Malone and Crowston 1994; McChesney and Gallagher 2004; Nurmi et al. 2005; Sutanto et al. 2004). Coordination and control have also been identified as central to the creation of organizational resilience in an IS industry context (Riolli and Savicki 2003). Hence, understanding these processes is key also to understanding DD as a resilient response to an ever changing business environment (see Lengnick-Hall 2005). In particular, the communication, coordination, and control activities are affected over a number of *dimensions*, which have been well

elaborated in the literature (e.g., Battin et al. 2001; Boland and Fitzgerald 2004; DeLone et al. 2005; Espinosa and Carmel 2003; Ghosh et al. 2004; Heeks et al. 2001; Sutanto et al. 2004). These relate to temporal, geographic and socio-cultural distance. These processes and dimensions have been incorporated into a framework of issues in distributed development (Ågerfalk et al. 2005).

    We will use this framework to present the results of our own study on strategies for effective DD, and so introduce it briefly here. Successful *communication* is "the exchange of complete and unambiguous information—that is, the sender and receiver can reach a common understanding" (Carmel and Agarwal 2001, p. 23). The communication process concerns the transfer of knowledge and information between actors, and the tools used to facilitate such interaction. *Coordination* is "the act of integrating each task with each organizational unit, so the unit contributes to the overall objective" (Carmel and Agarwal 2001, p. 23) The coordination process concerns how this inter-action makes actors interdependent on each other: "Two people have a coordination problem whenever they have common interests, or goals, and each person's actions depend on the actions of the other" (Clark 1996, p. 62). *Control* is "the process of adhering to goals, policies, standards, or quality levels" (Carmel and Agarwal 2001, p. 23). The control process concerns the management and reporting mechanisms put in place to make sure a development activity is progressing. *Temporal distance* is a directional measure of the dislocation in time experienced by two actors wishing to interact. Temporal distance can be caused by time zone difference or time shifting work patterns. In general, low temporal distance improves opportunities for timely synchronous communication but may reduce management options. *Geographical distance* is a directional measure of the effort required for one actor to visit another at the latter's home site. Geographical distance is best measured in ease of relocating rather than in kilometers. In general, low geographical distance offers greater scope for periods of collocated, inter-team working. *Socio-cultural distance* is a directional measure of an actor's understanding of another actor's values and normative practices. As a consequence, it is possible for actor A to be socio-culturally closer to actor B than B is to A. It is a complex dimension, involving organizational culture, national culture, language, politics, individual motivations, and work ethics. In general, low socio-cultural distance improves communication and lowers risk.

    A development context is considered distributed if it exhibits significant distance in the geographical dimension. We would consider a development team comprising members in two different offices in different cities within the same country to be distributed, even if they exhibit low temporal and socio-cultural distance. The key feature is that the cost (not necessarily monetary) to bring dispersed team members together is a significant inhibitor to spontaneous face-to-face meetings. When a DD project exhibits high distance in all dimensions, it is commonly referred to as a GSD project.

    The complete framework, presented as Table 1, forms a matrix in which each cell represents the impact of one dimension on one process. The table has been populated with an overview of the DD issues relating each process to each dimension (from Ågerfalk et al. 2005). This is the basis for our later analysis.

Table 1. An Overview of the Framework for Analyzing DD (adapted from Ågerfalk et al. 2005)

Process	Dimension		
	Temporal Distance	**Geographical Distance**	**Socio-Cultural Distance**
Communication	Reduced opportunities for synchronous communication, introducing delayed feedback. Improved record of communications.	Potential for closer proximity to market and utilization of remote skilled work forces. Increased cost and logistics of holding face to face meetings.	Potential for stimulating innovation and sharing best practice, but also for misunderstandings.
Coordination	With appropriate division of work, coordination needs can be minimized. Coordination costs typically increase with distance.	Increase in size and skills of labor pool can offer more flexible coordination planning. Reduced informal contact can lead to reduced trust and a lack of critical task awareness.	Potential for learning and access to richer skill set. Inconsistency in work practices can impinge on effective coordination, as can reduced cooperation through misunderstandings.
Control	Time zone effectiveness can be utilized for gaining efficient 24 × 7 work. Management of project artefacts may be subject to delays.	Difficult to convey vision and strategy. Communication channels often leave an audit trail, but can be threatened at key times.	Perceived threat from training low-cost rivals. Different perceptions of authority/hierarchy can undermine morale. Managers must adapt to local regulations.

3   RESEARCH METHOD

In conducting this research, our goal was to consider how companies may increase resilience through adopting effective DD practices. To this end, we have conducted a literature analysis with the aim of characterizing successful strategies for distributed development practice and relating these to the framework of Table 1.

We conducted an analysis of the published literature. For the literature analysis, systematic searches of the literature were made using keyword and author searches, and searches of tables of contents of journals and conference and workshop proceedings. Bibliographic databases were used to assist in forward and backward referencing. Papers were included if they had a core focus on DD, and were based on reported case or field studies in real industrial settings. An extensive note file was also compiled, with quoted sections from papers that contained their major import. This allowed faster filtering in the later stages of analysis, but context was always checked against the full text. The text resulting from this process was coded using a set of codes that evolved during the analysis. These codes form the resulting 10 strategies.

4   STRATEGIES USED IN SUCCESSFUL DD PROJECTS

In this section we consider the peer-reviewed literature on DD processes, specifically focusing on case studies and field studies in DD. The intention is to group and characterize the strategies proposed from real-world experience for reducing risk in DD and thereby leveraging its opportunities.

4.1 Have a Clear Distribution Rationale

When establishing a collaboration involving stakeholders with different native languages, there is a perceived increase in socio-cultural distance. For example, it has been argued that the "language factor is one of the reasons for the success of offshore IT work in countries with strong English language capabilities such as the Philippines and Singapore" (Carmel and Agarwal 2001, p. 27). An approach used by some U.S. companies is to "invest in English as a Foreign Language courses for those who are not fluent in English to improve professional communication" (Carmel and Agarwal 2001, p. 27).

In order to minimize the need for communication when identifying potentially successful development scenarios, Heeks et al. (2001) report that one should try to "focus on well-structured, stable projects"; this has "helped some case study clients push a lot of information exchange into the formal realm that IT-mediated distance can handle relatively well" (p. 59). Herbsleb and Grinter (1999b) elaborate on this idea, suggesting that, to the extent possible, one should "only split the development of well-understood products (or parts of products) where plans, processes and interfaces are established and likely to be stable" (p. 94). If stability is not achieved, the need for communication within the project will significantly increase.

Stability can be affected by socio-cultural distance regarding method usage. The issue of method transfer, even in non-distributed development contexts, has been shown to be a complex and difficult activity (Lings and Lundell 2004). Software development practice often involves improvisation and deviation from documented methods. To master the development processes used in a development project there is a need for informal communication, which for GSD implies travel and direct meetings between stakeholders (Heeks et al. 2001, p. 59).

In establishing an international project involving different sites, it is important to consider time-zone differences between sites. Minimizing time-zone differences facilitates effective synchronous communication, but eliminates the advantage of follow-the-sun type work (Carmel and Agarwal 2001, pp. 27-28). Consequently, establishing sites in a global project presents a trade-off with respect to temporal distance.

4.2 Clarify All Understandings

There are many informal agreements made between partners when setting up a distributed project, and these should be properly understood by all parties. One way in which to clarify is to document. The importance of documenting project goals is emphasized by Bass and Paulish (2004), based on studies at Siemens. They elaborate on the potential risks claiming that "in the absence of clear direction, local cultural and

personal biases are going to influence decisions. The resulting choices may not be in line with the overall goals of the project" (p. 10).

Based on GSD projects in the telecommunication company Alcatel, Ebert and DeNeve (2001) recommend clearly documenting all understandings. They suggest defining "at a project's beginning which teams are involved and what they will do in each location" and ensuring that "commitments exist in written and controlled form" (Ebert and DeNeve 2001, p. 68). It is particularly important to clarify understandings between teams in interorganizational collaborations. In an empirical study, Pyysiäinen (2003) found that background information was often lacking, causing problems in building trust between sites. It was found in the study that a "useful practice in the beginning of a project was a collocated training of the development process to be used" (p. 72).

It is also important to have mechanisms for monitoring goal fulfilment. This can be handled in different ways. For example, Boland and Fitzgerald (2004) report from a case study on GSD at Analog Devices that the software manager required each developer to "submit a task report at the beginning of each week," which helped to reduce inter-site dependencies. Such delivery reports contain

> a list of their specific goals for the week and a summary of their progress for the previous week. The report also indicates if the developer intends to make any deliveries during the week (i.e., check their work into the main source tree). This reporting process enables the software manager to be aware of work progressing across all the development sites and provides the necessary information to coordinate tasks among the developers (Boland and Fitzgerald 2004, p. 5).

However, they report that the strategy of using delivery reports was combined with strategies for temporary collocation (see section 4.8) for strengthening morale and motivation.

4.3 Leverage Modularity

The importance of a well-partitioned architecture is stressed by Bass and Paulish (2004), who claim that "in order to facilitate work break down across multiple sites, the architecture needed to reflect the organizational structure of the project" (p. 10). Based on their study at Siemens, they observed that there

> needed to be well-defined components or subsystems with understood dependencies for each site. These components or subsystems also needed to take into account the technical skills of the staff at the responsible development sites (p. 10).

In applying this strategy, the project itself may be made to reflect the structure of the system to be built, to guarantee no tension in the light of Conway's Law (which says that the structure of the system mirrors the structure of the organization that designed it). Herbsleb and Grinter (1999b) use this idea to recommend, "To the extent possible,

assign work to different sites according to the greatest possible architectural separation in a design that is as modular as possible" (p. 94).

At one extreme, the project may be broken down into multiple components at the start (Akmanligila and Palvia 2004). This is contrasted with an approach in which local requirements gathering is followed by collocation of representatives from each team for defining a common structure (see section 4.8).

Distributed component development brings with it the issue of system integration and the need to avoid a "big bang" integration activity within a project (Battin et al. 2001). From their case study at Motorola, they report that they "grew an incremental understanding of pair-wise network element interactions and never faced 'big bang' integration" (p. 73).

4.4 Use Cultural Mediation

Liaisons between teams have been found to be a very effective strategy for building trust in a project. For example, Battin et al. (2001) found, in a GSD project in Motorola, that liaisons were a good way for overcoming socio-cultural tensions within a project. In their own words, "The liaisons provided the key link between the architecture team and the development teams, as well as providing the US management team with a face to put with the non-US centers" (p. 74).

Herbsleb and Grinter (1999b) recommend that one creates a pool of liaisons in a project. Specifically, they recommend giving

> the early travelers the explicit assignment of meeting people in a variety of groups at the other site, and learning the overall organizational structure. Try to send gregarious people who will enjoy this role. When they return, make it known they can help with cross-site issues, and free up some of their time to do so (p. 94).

Many companies have project managers or key executives who act as cultural liaisons, implying that they frequently travel between the key stakeholder sites. In so doing, the role is "to facilitate the cultural, linguistic, and organizational flow of communication and to bridge cultures, mediate conflicts, and resolve cultural miscommunications" (Carmel and Agarwal 2001, p. 27). An interesting variation of this is put forward by Ebert and DeNeve (2001), based on experience from Alcatel, a large telecommunication company. They claim that management should rotate "across locations and cultures to create the necessary awareness for cultural diversity and how to cope with it" (Ebert and DeNeve 2001, p. 69).

Cultural mediation may also be facilitated by means of "straddlers" (Heeks et al. 2001) who bridge gaps in a project by having "one foot in the client's world and one in the developer's world" (p. 59). Effectively, straddlers are adept at bridging between two different development cultures having (usually) had experience in both.

The use of an offshore-onshore bridgehead in GSD is discussed by Carmel and Agarwal (2001), labeling this as

the 75/25 rule of thumb: Essentially, 75 percent of personnel work occurs offshore, while 25 percent occurs onshore (usually at the customer site—for example, in the US). This arrangement optimizes cost savings (offshore) while maintaining closeness to the customer. The individuals assigned to work onshore are typically the more experienced and culturally assimilated. They act to understand the customer's requirements specifications and translate them to the offshore programmers (p. 26).

Such an arrangement, by allowing the use of face-to-face communication, reduces miscommunication between stakeholders at different sites and has been found to be "reassuring" to customers (Carmel and Agarwal 2001, p. 26).

4.5 Facilitate Human Communication

Face-to-face communication is still acknowledged to be the best in most situations, but is clearly not always practical. Hence, a number of communication strategies have been used to maintain elements of synchronous communication. For example, Ebert and DeNeve (2001) recommend provision of "sufficient communication means, such as videoconferencing or shared workspaces and global software libraries" (p. 69) as an approach for improving human communication within distributed projects. However, current technology often brings with it the inherent "challenge of delay due to inadequate (asynchronous) communication" (Damian and Zowghi 2002, p. 10).

Battin et al. (2001) report that they met their "real-time communications needs by teleconferencing," which "became a critical component" in their communication strategy (p. 72). Interestingly, they used conference calls despite the fact that they could "report a problem by email almost instantaneously to all teams," reasoning that "resolution often required detailed discussions" (p. 72). To overcome time-zone problems in arranging such meetings they schedule discussions "during the night from the site requesting the conference call" (p. 72).

In reporting from a field study conducted in a multisite organization, distributed over five continents, Damian and Zowghi (2002) discuss strategies for improving informal human communication among team members through initial face-to-face kick-off meetings (which relates to the strategy of temporary collocation; see section 4.8), and "on-going scheduled informal meetings across sites" (p. 9). Electronically equipped rooms were provided for "drop-in" purposes "to share work artifacts as they would if they started a design discussion near someone's cubicle" (Damian and Zowghi 2002, p. 10). The usefulness of such chat between developers in problem solving situations was also identified by Paasivara (2003, p. 62). In the study, Paasivara notes that developers felt that when chatting they were able to easily post "clarifying counter questions" and that "chat session can be open all the time" (p. 62).

4.6 Manage Processes

From their study at Siemens, Bass and Paulish (2004, p. 10) note the importance of weekly teleconferences to monitor status and highlight issues. They stress the

importance (but acknowledge the difficulty) of taking into account time zones and local holiday schedules when scheduling such meetings. When sites have some overlapping time, it is good to plan the work process at each site so that overlap time can be devoted to such meetings (Espinosa and Carmel 2003). Such time can be increased by modifying work patterns. Paasivara (2003) observes that weekly meetings are appropriate for information and monitoring purposes in both directions (i.e., customer to supplier and vice versa), and recommends an agenda that concentrates on "tasks done, tasks to be done, problems and open issues" (p. 62).

Leadership is important for managing software development processes, and perhaps even more so when managing distributed projects. Ebert and DeNeve (2001, p. 68) recommend that a project should have "one project leader who is fully responsible for achieving project targets," and that members of the project management team should represent "the major cultures within the project" (p. 68).

Based on a field-study, Passivara and Lassenius (2004) report that design and code reviews "seemed to be useful in distributed projects with distant sites or subcontractors" (p. 44). They note that such "reviews are early checks that the distributed teams have understood the requirements correctly and are doing what they are supposed to do" (p. 44).

4.7 Develop a Sense of "Teamness"

To strengthen the team culture, Ebert and DeNeve (2001) recommend setting up "a project homepage that summarizes project content, progress metrics, planning information, and team-specific information" (pp. 68-69). Bass and Paulish (2004) note the importance of such measures for communicating progress to team members. They report from a study in Siemens how making the URL for a test system available for all the team members boosted morale for the team: members became aware of the rapid progress being made. "The result was a much greater sense of team than would otherwise have been possible in a globally distributed project" (Bass and Paulish 2004, p. 10).

On the content of a common web site, Espinosa and Carmel (2003) recommend various awareness tactics related to time and work hours, including publishing hours and time differences for the different sites. Damian and Zowghi (2002) recommend going beyond a simple home page to the use of "collaborative Internet technologies" for synchronous testing and collaborative prototyping activities (p. 9). They suggest the use of a human facilitator and "an integrated, richer communication media that integrates data, video and audio channels, in the decision-making teleconferencing calls" (p. 10). Using such an approach in an intercontinental project, they perceived more effective requirements decision-making meetings and improved conflict management.

The issue of trust is closely related to encouragement of a team culture in a project. As noted by Pyysiäinen (2003), properly informing all stakeholders about project progress is also important for strengthening trust in a team. Instead of quantitative feedback (number of working hours etc.), it is important to provide feedback on "quality and concrete contributions of the deliverables" (p. 73). However, perceptions of trust can vary. For example, Damian and Zowghi noted clear differences between how Australian and American stakeholders perceived the importance of *trust*. In their own words,

while "trust" was a word often heard in the interviews with the Australian group, for the American stakeholders trust was not an issue. While it is clear that this is due to some sort of cultural difference, one may believe that it is a matter of national or functional culture differences (p. 4).

Part of building a team culture is to reduce the socio-cultural distance between stakeholders within a firm. To this end, Carmel and Agarwal (2001) note the strategy of establishing software centers in other countries rather than outsourcing, bringing IT workers "within the corporate network—inside the firewall—with access to all knowledge-bases, calendars, Web pages, and so forth. They are also trained in the corporate methodologies, policies, and systems" (p. 26).

4.8 Encourage Temporary Collocation

When companies undertake parallel development activities, they sometimes temporarily collocate people. Such meetings are often used to synchronize activity, but may also be used to strengthen morale and lower socio-cultural distance.

Boland and Fitzgerald (2004) report on the use of quarterly sync-up meetings as a very successful strategy for maintaining morale and motivation among team members. They observed that among developers there were comments on "feeling 'energized' and highly motivated after meetings with all the team members" (p. 6) Heeks et al. (2001) report on extensive use of such meetings which "proved to be more effective at synching values and informal information, in a way that IT-mediated communication could not" (p. 56). Visits were undertaken both ways (i.e., North America to India and vice versa).

Temporary collocation is also recommended by Damian and Zowghi (2002) as an approach for improving "awareness of users' local working context" and for contributing to "better communication with sources of requirements through a more appropriate participation from field personnel" (p. 9). It can also be used to strengthen the liaison role in cultural mediation (see section 4.4). Espinosa and Carmel (2003) report, from experiences of UK, German, and Indian software teams, that it is common for Indian team members to be trained in the UK and Germany for a few months. Thereafter, they go back to India and "serve as points of contact for the UK and German developers" (p. 252).

With respect to scheduling periods of collocation, it is recommended to front load travel in a project. Pyysiäinen (2003) notes that a "common kick-off meeting" in the beginning of the project was found to be a "successful way to create initial familiarity between members" (p. 72). Herbsleb and Grinter (1999b) put it this way: "bring people who need to communicate together early on. All other means of communication will work better once developers, testers, and managers have some face-to-face time together" (p. 94).

4.9 Encompass Heterogeneity

It may be that homogeneity appears attractive within a distributed project, but heterogeneity is likely to be unavoidable and so should be carefully planned for. There may be heterogeneity in methods and/or tools and/or terminology.

Battin et al. (2001) report on the need to accommodate existing processes, to "let each team begin producing results immediately, using a process they were familiar with. If the teams had been forced into a common process, the learning curve would have impacted the delivery of the system" (p. 75).

To cater for heterogeneity in process Ebert and DeNeve (2001) recommend providing "an interactive process model based on accepted best practices that allows tailoring processes for the specific needs of a project or even team" (p. 69).

A related problem concerns notations and terminology used in a project. This was experienced in the project analyzed by Battin et al. (200, p. 75): "We understood the inconsistency in notations and terminology in the beginning of the project and came up with a set of common 'work products' and vocabulary." They emphasize the need for standardization in documentation at the project level to facilitate tracking in the shared project databases.

Although potentially advantageous, homogeneity may not be achievable in the tools chosen for a project. For example, the same version of a tool may not be marketed and supported in all locations. As experienced by Battin et al. (2001, p. 74), "Obtaining the same version of a product from multiple sales teams proved quite difficult. While the latest version of most products was readily available in the US, the vendors were often still introducing previous versions in other countries."

Given this, it might be tempting to consider shipping a common tool set to all sites. Apart from ensuing support problems, export licences may not be available. The use of tools under an Open Source licence would naturally change the nature of this problem.

4.10 Develop an Effective Tool Base

Battin et al. (2001, p. 74) recommend the adoption of a common SCM tool and problem tracking tool for all sites. With respect to tools, they note that it is "less important to focus on the particular tools" than to understanding the functions these tools support. This is also emphasized by Herbsleb and Grinter (1999a), who recommend that one invest in "tools that address the real problems" (p. 70). By this, they mean tools that "make it easier to find organizational information, to maintain awareness about the availability of people, and to have more effective cross-site meetings, especially spontaneous ad hoc sessions" (p. 70).

To handle time separation between developers in a distributed project, a number of support tools may be used. A key for achieving this is to

> make better use of asynchronous technologies, such as electronic mail, voice mail, and use of various shared databases and other repositories (groupware, knowledge management, team intranets and web sites, discussion areas, etc.) (Espinosa and Carmel 2003, p. 251).

However, Herbsleb and Grinter (1999a) point out that although "video conferencing, desktop video, electronic bulletin boards, and workflow applications might add value in some circumstances," such tools "do not directly address the core problems" (p. 70).

5   ANALYSIS OF STRATEGIES FOR DD SUCCESS

In this section, we summarize the strategies for DD success, and position them within the framework of Table 1. In so doing, the 10 strategies are related to opportunities and threats in DD. We then consider practitioner literature, as a check for congruence with the peer-reviewed research sources. The 10 strategies for DD success are first summarized and then related to the framework of Table 1.

5.1 A Summary of the Ten Strategies

S1: Have a clear distribution rationale: Not all projects and not all collaboration contexts are equally amenable to DD. From a context perspective, choose offshore teams with a language in common. It may be advantageous to select for low temporal distance, unless follow-the-sun working is relevant. In any case, guarantee regular working time overlap between sites. Rigorously enforce an acceptable capability maturity level of all partners. From a project perspective, only consider DD for well structured, well understood and stable projects, decomposable into discrete tasks.

S2: Clarify all understandings: At the start of any project agree and communicate project goals and targets, and ensure that commitments are genuinely understood. Define which teams are involved, and what will be done in each location. Further, agree and document binding interorganizational processes and stabilizing processes.

S3: Leverage modularity: A system architecture mirrors the structure of the organization that built it (Conway's law), so for software development work, plan the architecture of the system around the distributed structure of the team. This will reduce the need for intensive collaboration, and allow optimum utilization of local skills. For other life-cycle phases plan natural divisions of work in relatively small bundles.

S4: Use cultural mediation: Training in cultural issues is useful. Beyond that, use a cultural mediator, or liaison. This is a person from one team context spending time in another, and becoming a link person between the teams. Many GSD teams use liaisons, who may spend short periods relocated or may even be relocated for an entire project—effectively becoming part of a bridgehead. A more radical suggestion is to rotate management across locations (and therefore cultures) to improve awareness.

S5: Facilitate human communication: Synchronous communication is most effective face to face, but a number of strategies can address the weaknesses of remote communication. Providing rich technologies may help, but improving efficacy of standard technologies is important. A human facilitator in teleconferencing can reduce misunderstandings and smooth conflicts. Language classes can improve confidence and reduce a tendency to asynchronous forms of communication. Increasing informal communication and past face to face meetings can lead to improvements in more formal meetings.

S6: Manage processes: Having one, identified project leader with full responsibility should be supplemented with team and local project managers, even though responsibilities overlap. Regular teleconferences and regular developer reports are recommended for monitoring project status. Plan meetings to occur during overlapping working hours, which can be expanded by time-shifting. Synchronizing

delivery and integration cycles between partners, and instigating design and code reviews to verify requirements, are important. Incremental development and release schedules with short cycles are also cited.

   S7: Develop a sense of teamness: Common strategies include the development of a project home page, which includes team member details and important planning information such as national holidays. Also summarize project progress as well as planning and team-specific information. Record decisions and make them easily accessible. Ensure timely feedback to communications about progress, including deliverables. Real-time sharing of artefacts, including ideas, perhaps further facilitated by time-shifting.

   S8: Encourage temporary collocation: Investing in periods of collocation for teams can reduce future problems in all future processes, but such relocations need planning and can be expensive. Consider collocating developers, not only managers. There may be a one-off project initiation session, where understandings are forged and strategic thinking can take place. There may also be regular (e.g., quarterly) synchronization and review meetings, but front-loading travel is considered most effective. Variation includes project phasing, with one phase distributed and another phase in-house.

   S9: Encompass heterogeneity: There can be advantages in accommodating heterogeneous methods, tools, and terminology, but such accommodation needs to be planned and catered for. Tool heterogeneity may be forced because of local restrictions (export licensing, available support, etc.). Local terms and concepts need to be mapped to a common ontology to prevent project-level confusion. One suggested strategy is to provide an interactive process model that can be tailored for each team.

   S10: Develop an effective tool base: A common software configuration management tool is recommended for coordination, probably replicated at each site. This can be enhanced by creative use of the comments fields as an extra form of asynchronous communication. The key thing is to invest in tools that address the real problems. Tool take-up is otherwise low.

5.2 Relating the Strategies to the Framework

   The first strategy—have a clear distribution rationale—addresses primarily problems associated with temporal distance by reducing the need for communication, which in turn simplifies coordination and control. Reducing communication also reduces potential problems in the socio-cultural dimension, such as culturally induced misunderstandings.

   The second strategy—clarify all understanding—is mainly a way to minimize potential misunderstandings and communication breakdowns which can result from non-overlapping socio-cultural backgrounds. Clearly documenting such things as project goals and individual partner commitments helps to remove the communication problems otherwise caused through differing interpretations of informal agreements.

   The third strategy—leverage modularity—suggests that the system architecture should be designed to reflect the geographical (and competence) structure of the project. In this way, local expertise can be utilized efficiently, thus reducing potential coordination and control problems.

The fourth strategy—use cultural mediation—suggests that it is worth spending resources on reducing socio-cultural distance by means of facilitating face-to-face meetings. Different approaches can be used, but the main idea is to have at least some people at each node who have met people at peer nodes in person. This also reduces the perceived geographical distance, if not the physical.

The fifth strategy—facilitate human communication—focuses on the communication process across all three dimensions of distance. Good communication is also fundamental for successful coordination and control and so can indirectly be seen to address these processes also. The utilization of innovative IT-based solutions for real-time conversations is crucial for succeeding with this strategy.

The sixth strategy—manage processes—addresses the control and coordination structure of a project with respect to temporal distance. Basically, there need to be processes in place for harmonizing tasks between nodes at predefined points in time, so that all nodes can plan their work around these contact points.

The seventh strategy—develop a sense of teamness—aims to facilitate communication and coordination by stimulating the feeling of being a member of a team. A project is more likely to be successful when all members share a sense of belonging to the same team.

The eighth strategy—encourage temporary collocation—takes the cultural mediation strategy even further by suggesting that all developers should spend time at remote sites on a temporary basis. If a cultural liaison facilitates communication between sites, having local peers at the remote site more directly increases the team's coordination.

The ninth strategy—encompass heterogeneity—aims to prepare for problems introduced by the fact that any DD team is naturally heterogeneous. Allowing for different work practices but managing these through a common method tailoring framework is central to successful coordination and control.

The tenth and final strategy—develop an effective tool base—aims to facilitate coordination and control through the use of standardized tool support for configuration and change management.

The 10 strategies and how they map to the framework of Table 1 are shown in Table 2. Each of the 10 strategies has been positioned in Table 2 according to its main emphases.

From Table 2 we can conclude that there are indeed DD strategies that address all problem areas constituted by the nine cells of the framework. However, this does not mean that all problems are solved. It may be tempting to think of Table 2 as a tool to find an optimal minimal set of strategies that will cover all nine DD problem areas. This is not advisable since there is no guarantee that any one strategy is either necessary or sufficient to overcome problems in any particular area. Rather, the mapping should be seen as a guide to which areas may have been left out should particular strategies not have been put into practice. It is also a fact that the success of each strategy is contingent upon the particular organizational context and so must be tailored to suit each specific situation.

The fact that most of the strategies deal with the socio-cultural dimension could be interpreted in two quite different ways. On the one hand, it could mean that this dimension is particularly problematic and important, hence a lot of effort has been spent on reducing socio-cultural distance. On the other hand, it could mean that this dimension is trivial and that many obvious strategies have emerged. Judging by the many problems reported in the literature, the former is probably the most likely.

Table 2. Positioning the Strategies for DD Success Within the Framework of Table 1

Process	Temporal Distance	Dimension	
		Geographical Distance	**Socio-Cultural Distance**
Communication	Have a clear distribution rationale (S1) Facilitate human communication (S5)	Use cultural mediation (S4) Facilitate human communication (S5) Encourage temporary collocation (S8)	Have a clear distribution rationale (S1) Clarify all understandings (S2) Use cultural mediation (S4) Develop a sense of teamness (S7) Encourage temporary collocation (S8)
Coordination	Have a clear distribution rationale (S1) Manage processes (S6) Develop an effective tool base (S10)	Leverage modularity (S3) Encourage temporary collocation (S8) Develop an effective tool base (S10)	Clarify all understandings (S2) Develop a sense of teamness (S7) Encourage temporary collocation (S8) Encompass heterogeneity (S9)
Control	Have a clear distribution rationale (S1) Manage processes (S6) Develop an effective tool base (S10)	Leverage modularity (S3) Develop an effective tool base (S10)	Clarify all understandings (S2) Encompass heterogeneity (S9)

5.3 Congruence with Practitioner Viewpoints

The practitioner literature is largely consistent with the research literature, acknowledging the problems and dimensions of DD (see, for example, Coar 2003-2004), but also giving some pragmatic insights into experience of DD. For example, the increased risks are well recognized, including that associated with the unsettling and potentially demotivating effects of major outsourcing decisions (Goulston 2004). However, the need for CIOs to be proactively following the lead of large corporations in outsourcing is seen as a driving force for increased globalization at least over the medium term (Smith 2004). Practitioner guidelines are largely consistent with the strategies outlined above, although some detail is added. For example, Smith goes on to detail a *collocation* strategy of keeping prototyping and piloting work in-house but outsourcing production. Turnlund (2003-2004) emphasizes the importance of leveraging modularity in his "workgroup containment" rule. The general consensus seems to be that outsourcing "means trouble for the unprepared" (Grossman 2003).

6   CONCLUSIONS

In this paper we have considered how companies may become more resilient through adopting effective DD practices. Since DD could be seen both as a response to external pressures and as a disruptive innovation that may well introduce new internal turbulence, understanding the particular DD challenges and opportunities is crucial for any organization adopting DD. In order to adapt to changing circumstances brought about by DD, successful strategies for coping with the processes of coordination, control, and communication must be adopted. To understand these processes in the context of DD, we have used a framework that combines the three processes with the three distances characterizing DD: temporal distance, geographical distance and socio-cultural distance. Altogether this framework thus provides nine areas that pose challenges and opportunities for DD projects. Based on existing literature from case and field studies we have synthesized and presented ten general strategies for successful DD. These strategies have been shown to address all of the nine DD problem areas of the framework—albeit the extent to which they can be combined to synergistically solve all major DD problems remains as a future research topic. Consequently, further deep case and field studies are needed.

Although we have considered only traditional DD in this study, there are many striking examples of successful distributed development in the area of open source systems development. Some even conjecture that paradigms encompassing the successful strategies of both OSS and commercial projects are the holy grail of distributed development, enabling cross-fertilization of ideas throughout traditional distributed and OSS development. Such studies would allow further development of the strategies presented here, with a view to informing best practice throughout DD, and thereby increasing resilience in software development companies.

Acknowledgments

References

Ågerfalk, P., Fitzgerald, B., Holmström, H., Lings, B., Lundell, B., and Ó Conchúir, E. "Framework for Considering Opportunities and Threats in Distributed Software Development," in *Proceedings of the International Workshop on Distributed Software Engineering*, Austrian Computer Society, 2005, p. 47-61.
Akmanligil, M., and Palvia, P. C. "Strategies for Global Information Systems Development," *Information & Management* (42:1), 2004, pp. 45-59.
Bass, M., and Paulish, D. "Global Software Development Process Research at Siemens," in *Proceedings of the 3rd International Workshop on Global Software Development* (collocated with ICSE 2004, International Conference on Software Engineering), Edinburgh, Scotland, May 24, 2004, pp. 8-11 (available online at http://gsd2004.cs.uvic.ca/docs/proceedings.pdf).

Battin, R .D., Crocker, R., Kreidler, J., and Subramanian, K. "Leveraging Resources in Global Software Development," *IEEE Software* (18:2), 2001, pp. 70-77.

Boland, D., and Fitzgerald, B. "Transitioning from a Co-Located to a Globally-Distributed Software Development Team: A Case Study at Analog Devices Inc.," in *Proceedings of the 3rd International Workshop on Global Software Development*, (co-located with ICSE 2004, International Conference on Software Engineering), Edinburgh, Scotland, May 24, 2004, pp. 4-7 (available online at http://gsd2004.cs.uvic.ca/docs/proceedings.pdf).

Brooks, F. P. Jr. "No Silver Bullet: Essence and Accidents of Software Engineering," in H. J. Kugler (ed.), *Information Processing 1986*, Amsterdam: Elsevier Science Publishers B.V. (North-Holland), 1986, pp. 1069-1076.

Carmel, E. "Introduction to the Special Issue of EJISD: The Emergence of Software Exporting Industries in Dozens of Developing and Emerging Economies," *The Electronic Journal on Information Systems in Developing Countries* (13), Special Issue, May 2003, pp. 1-2 (available online at www.ejisdc.org).

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

Clark, H. H. *Using Language*, Cambridge, England: Cambridge University Press, 1996.

Coar, K. "The Sun Never Sets on Distributed Development," *Queue*, December/January 2003-2004, pp. 32-39.

Damian, D., Lanubile, F., and Oppenheimer, H. L. "Addressing the Challenges of Software Industry Globalization: The Workshop on Global Software Development," in *Proceedings 25th International Conference on Software Engineering*, Los Alamitos, CA: IEEE Computer Society Press, 2003, pp. 793-794.

Damian, D. E., and Zowghi, D. "The Impact of Stakeholders' Geographical Distribution on Managing Requirements in a Multisite Organization," in *Proceedings IEEE Joint International Conference on Requirements Engineering*, Los Alamitos, CA: IEEE Computer Society Press, 2002, pp. 319-328.

DeLone, W., Espinosa, J. A., Lee, G., and Carmel, E. "Bridging Global Boundaries for IS Project Success," in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 1*, Los Alamitos, CA: IEEE Computer Society Press, 2005, pp. 1-10.

Ebert, C., and De Neve, P. "Surviving Global Software Development," *IEEE Software* (18:2), 2001, pp. 62-69.

Espinosa, A., and Carmel, E. "The Impact of Time Separation on Coordination in Global Software Teams: A Conceptual Foundation," *Software Process Improvement and Practice* (8), 2003, pp. 249-266.

Evaristo, J. R., Scudder, R., Desouza, K. C., and Sato, O. "A Dimensional Analysis of Geographically Distributed Project Teams: A Case Study," *Journal of Engineering and Technology Management* (21:3), 2004, pp. 175-189.

Goulston, M. "The Inner Cost of Outsourcing: When Contemplating Outsourcing, CIOs Should First Think About Their People," *CIO Magazine*, November 1, 2004 (available online at www.cio.com/archive/110104/interview.html).

Ghosh, T., Yates, J. A., and Orlikowski, W. J. "Using Communication Norms for Coordination: Evidence from a Distributed Team," in R. Agarwal, L. Kirsch, and J. I. DeGross (eds.), *Proceedings of the 25th International Conference on Information Systems*, Washington, DC, December 2005, pp. 115-127.

Grossman, E. (ed.). "New World Order," *Queue*, December/January 2003-2004, pp. 27-31.

Heeks, R., Krishna, S., Nicholson, B., and Sahay, S. "Synching or Sinking: Global Software Outsourcing Relationships," *IEEE Software* (18:2), 2001, pp. 54-60.

Herbsleb, J. D., and Grinter, R. E. "Architectures, Coordination, and Distance: Conway's Law and Beyond," *IEEE Software* (16:5), 1999a, pp. 63-70.

Herbsleb, J. D., and Grinter, R. E. "Splitting the Organization and Integrating the Code: Conway's Law Revisited," in *Proceedings of the 21tst International Conference on Software Engineering (ICSE'99)*, New York: ACM Press, 1999b, pp. 85-95.

Lengnick-Hall, C. A. "Adaptive Fit Versus Robust Transformation: How Organizations Respond to Environmental Change," *Journal of Management* (31:5), 2005, pp. 738-757.

Lings, B., and Lundell, B. "On Transferring a Method into a Usage Situation," in B. Kaplan, D. P. Truex III, D. Wastell, A. T. Wood-Harper, and J. I. DeGross (eds.), *Information Systems Research: Relevant Theory and Informed Practice*, Boston: Kluwer, 2004, pp. 535-553.

Lyytinen, K., and Rose, G. M. "The Disruptive Nature of Information Technology Innovations: The Case of Internet Computing in Systems Development Organizations," *MIS Quarterly* (27:4), 2003, pp. 557-595.

Malone, T. W., and Crowston, K. "The Interdisciplinary Study of Coordination," *ACM Computing Surveys* (26:1), 1994, pp. 87-119.

McChesney, I. R., and Gallagher, S. "Communication and Co-ordination Practices in Software Engineering Projects," *Information and Software Technology* (46:7), 2004, pp. 473-489.

Nurmi, A., Hallikainen, P., and Rossi, M. "Coordination of Outsourced Information System Development in Multiple Customer Environment: A Case Study of a Joint Information System Development Project," in *Proceedings of the 38th Hawaii International Conference on System Sciences*, Los Alamitos, CA: IEEE Computer Society Press, 2005, pp. 1-10.

Paasivaara, M. "Communication Needs, Practices and Supporting Structures in Global Inter-Organizational Software Development Projects," in *Proceedings of International Workshop on Global Software Development* (co-located with ICSE 2003, International Conference on Software Engineering), Portland, Oregon, May 9, 2003, pp. 59-63 (available online at http://gsd2003.cs.uvic.ca/gsd2003proceedings.pdf).

Paasivaara, M., and Lassenius, C. "Using Research Methodologies and Challenges in GSD," in *Proceedings of the 3rd International Workshop on Global Software Development* (co-located with ICSE 2004, International Conference on Software Engineering), Edinburgh, Scotland, May 24, 2004, pp. 42-47 (available online at http://gsd2003.cs.uvic.ca/gsd2003proceedings.pdf).

Pyysiäinen, J. "Building Trust in Global Inter-Organizational Software Development Projects: Problems and Practices," in *International Workshop on Global Software Development* (co-located with ICSE 2003, International Conference on Software Engineering), Portland, Oregon, May 9, 2003, pp. 69-74 (available online at gsd2003.cs.uvic.ca/gsd2003proceedings.pdf).

Riolli, L., and Savicki, V. "Information System Organizational Resilience," *Omega: The International Journal of Management Science* (31), 2003, pp. 227-233.

Smith, G. "You Can't Outsource Everything," *CIO Magazine*, November 1, 2004 (available online at http://www.cio.com/archive/110104/peer.html).

Sutanto, J., Kankanhalli, A., and Tan, B. C. Y. "Task Coordination in Global Virtual Teams," in R. Agarwal, L. Kirsch, and J. I. DeGross (eds.), *Proceedings of the 25th International Conference on Information Systems*, Washington, DC, December 2004, pp. 807-819.

Turnlund, M. "Distributed Development Lessons: Why Repeat the Mistakes of the past If You Don't Have To?," *Queue*, December/January 2003-2004, pp. 27-31.

United Nations. *World Investment Report 2004: The Shift Towards Services*, New York: United Nations Conference on Trade and Development, 2004.

About the Authors

Brian Lings, after a number of years at the University of Queensland, Australia, joined the Department of Computer Science at the University of Exeter, becoming its first elected head of

department. He is a consultant with Certus Technology Associates and a member of the academic staff of the University of Skövde, Sweden. He chairs the steering group of the BNCOD database conference, the main forum for database researchers in the UK. His research centers on the socio-technical evaluation of tool and method support for model-based distributed development. He is a codeveloper of the *2G* method, and continues to be active in applications of the method. Brian can be reached by e-mail at brian.lings@his.se.

Björn Lundell has been a staff member at the University of Skövde since 1984. He has contributed to international standardization (ISO), established active links with a number of Swedish private and public organizations, and coleads the work package on distributed development in the EU FP6 Co-ordination Action project CALIBRE (www.calibre.ie). He is a co-developer of the *2G* method, a qualitative method evolved for use in socio-technical evaluations which has been applied to the analysis of open source software usage in organizations. Over the past 5 years, he has gained significant experience in conducting case studies for analyzing and evaluating socio-technical phenomena in organizational contexts. His research is published in a variety of international journals and conferences. He has a general interest in qualitative methods and his research centers on the issues of technology evaluation, open source and distributed development models, and theoretical and practical aspects of method transfer into real organizational usage. Björn can be reached by e-mail at bjorn.lundell@his.se.

Pär J Ågerfalk received his Ph.D. from Linköping University and is currently a postdoctoral research fellow at the University of Limerick and an assistant professor at Örebro University. He is the deputy coordinator and scientific manager of the EU FP6 project CALIBRE and also co-leader of the work package on distributed development. His current research centers on open source software development in the secondary software sector, globally distributed and flexible software development methods, and how information systems development approaches can be informed by language/action theory. His over 40 peer-reviewed publications have appeared in a variety of international journals, books, and conference proceedings, and he is currently an associate editor of *European Journal of Information Systems* and a guest editor of *Communications of the ACM* and *Software Process: Improvement and Practice*. Pär can be reached by e-mail at par.agerfalk@ul.ie.

Brian Fitzgerald works at the University of Limerick, Ireland, where he is a research fellow and Science Foundation Ireland Principal Investigator. He has a Ph.D. from the University of London and has held visiting positions in Sweden, the United Kingdom, and United States. His publications include seven books and more than 80 papers, published in leading international conferences and journals. Brian has attracted research funding in excess of 10 million euro overall on his projects. Having worked in industry prior to taking up an academic position, he has more than 20 years experience in the software field. This experience was gained in a range of sectors and in several countries including Ireland, the UK, Belgium, and Germany. Brian can be reached by e-mail at brian.fitzgerald@ul.ie.