ULRR

A reference model for successful distributed development of software systems

Item Type	Meetings and Proceedings
Authors	Lings, Brian;Lundell, Bjorn;Agerfalk, Par J.;Fitzgerald, Brian
Citation	2nd International Conference on Global Software Engineering;2007
Publisher	IEEE Computer Society
Download date	2026-03-14 22:58:11
Item License	https://creativecommons.org/licenses/by-nc-sa/1.0/
Link to Item	https://hdl.handle.net/10344/2130

A reference model for successful Distributed Development of Software Systems

Brian Lings[1], Björn Lundell[1], Pär J. Ågerfalk[2,3], and Brian Fitzgerald[2]
1: University of Skövde, Sweden
2: Lero – University of Limerick, Ireland
3: Uppsala University, Sweden
brian.lings@his.se; bjorn.lundell@his.se; par.agerfalk@ul.ie; brian.fitzgerald@ul.ie

Abstract

Distributed Development (DD) of Software Systems is an issue of increasing significance for organisations today, all the more so given the current trend towards globalisation. In this paper we present a reference model which can be used as a reference point for any company wishing to review their own DD scenario. This is particularised in two forms, one as an exemplar model for a global (GSD) development scenario and one as a particularisation of this for intra-national DD scenarios. By drawing from eight case-studies on DD, we present ten general strategies for successful DD together with our reference model which characterises an ideal DD situation.

1. Introduction

The core challenges of distributed development (DD) of software systems seem to lie in the complexity of maintaining good communication, coordination and control when teams are dispersed in time (e.g. across time zones) and space, as well as socio-culturally. However, proven methods for successful DD have not yet been formulated, and there is a need for a better understanding of the process dimension of DD.

The issue of DD is a complex one, and has been categorised in a number of ways (e.g. [1] [2]). Much of the previous research on DD has concentrated on inter-continental DD (also referred to as GSD), even though a number of companies nowadays are involved in intra-continental and intra-national DD which also implies certain complexity in collaborations between sites. In fact, research has shown that even when the geographical spread of sites is very limited (for example when stakeholders do not naturally share a coffee room), the complexity in such DD needs consideration (e.g. [23]).

By drawing from the case- and field-study literature on DD, and the results from eight case-studies on DD, we present ten general strategies for successful DD together with a reference model which characterises an ideal situation for DD. The reference model is not intended as a blueprint for DD in any organisational context. What is intended is that it should be used to ask the correct questions when setting up a DD activity. The intention is that any inconsistencies with the reference model should be justified in the particular context.

We particularise the reference model for a global (GSD) development scenario and also for an intra-national DD scenario.

2. Research Approach

We have earlier [27] articulated ten strategies for successful DD practice which evolved from an analysis of case- and field-studies reported in the literature. In this paper we report from a deeper analysis which leads to a richer characterisation and justification of those ten strategies. In effect, the strategies represent a basic reference set of strategies for successful DD practice.

It is a fact that the success of each strategy is contingent upon the particular organisational context in which it is used, and so must be tailored to suit each specific situation. Consequently, we have investigated DD practice in DD contexts of 8 companies representing a wide range of company sizes.

Our first set of case studies targeted global software development (GSD), in which temporal, geographical and socio-cultural distances [1] were all high. These studies focused on three large companies, one being a semi-conductor company, one a financial investment company, and one an IT-manufacturer. Each company has headquarters in the USA with development teams within Europe. All three European sites coordinate with their remote colleagues in the USA and with

International Conference on Global Software Engineering(ICGSE 2007)
0-7695-2920-8/07 $25.00 © 2007 IEEE

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: University of Limerick. Downloaded on May 12, 2009 at 04:33 from IEEE Xplore. Restrictions apply.

others, for example in India, Poland and Malaysia. For these, all interviews were conducted at the European sites.

We supplemented these with studies of five companies in which temporal and socio-cultural distances were low, and geographic distance the primary issue. Case studies were conducted in two large companies: a car manufacturing company and a telecom company. These were supplemented with studies in three SMEs. Together, these studies give insight into DD within an intra-European context.

Analysis within the case studies initially concentrate on distance in DD along the three distance dimensions (see section 3), as these are the primary indicators of GSD. This also makes it possible to use these case studies, together with other intra-European studies, to investigate the factors primarily related to geographical rather than temporal or socio-cultural distance. This is important in considering the intra-European situation.

Data from each case study was then coded using the empirically grounded strategies evolved from the literature study of previous case- and field-studies in DD. The coded data was then further analysed to provide succinct characterisations of each case. These were then systematically integrated and refined into a reference model for DD, which we particularised for GSD. The idea of a reference model is not that it should be used in all situations; rather it can be used to reason about best practice for a given situation. Deviations from the model should be justified, for example by arguing that distance is reduced in the given context for one or more of the DD dimensions.

One such scenario was considered to be of particular interest in Europe, so a specialised reference model was considered. By analysing the geographical dimension only, and combining further coded data from the other studies, the general reference model was particularised to evolve a reference model for intra-national DD. This is ongoing work, and the results reported here should be considered initial.

Consideration was finally given to open source software development processes as a DD practice. In this case, 'organisations' are much less structured and so the process (communication, coordination and control) aspects of the framework were used to gain insight into the processes involved. An analysis was thereby conducted of strategies for open source software success.

3. On Distributed Development of Software Systems

In any software life-cycle process, a number of different kinds of activity are actually undertaken to facilitate the efficient use of multiple developers. These activities can be characterised broadly as communication, coordination and control activities.

In essence, communication is "the exchange of complete and unambiguous information – that is, the sender and receiver can reach a common understanding." [8] 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 organisational unit, so the unit contributes to the overall objective." [8] The coordination process concerns how this interaction 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." [11, p. 62] Control is "the process of adhering to goals, policies, standards, or quality levels." [8] The control process concerns the management and reporting mechanisms put in place to make sure a development activity is progressing.

In a context of distributed development of software systems, we recognise that there can be different kinds of distribution, giving rise to temporal, geographical and socio-cultural distances between developers [1]. We characterise each kind of distance in turn. 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 kilometres. In general, low geographical distance offers greater scope for periods of co-located, 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 organisational culture, national culture and language, politics, and 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

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: University of Limerick. Downloaded on May 12, 2009 at 04:33 from IEEE Xplore. Restrictions apply.

(not necessarily monetary) to bring dispersed team members together is a significant inhibiter 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 annotated framework [1, 28] forms a matrix in which each cell represents the impact of one dimension on one process.

We have analysed the peer reviewed literature on DD processes, specifically focusing on case studies and field studies in DD. Our intention has been to group and characterise the strategies proposed from real-world experience for reducing risk in DD and thereby leveraging its opportunities. Our focus has been to cover those peer-reviewed research studies which draw from practice of traditional distributed development. The goal has been to inform the deeper analysis of strategies for successful DD presented in the form of a reference model in section 5. Overall, in addition to our own case studies (see section 4), we have analysed a large set of published peer-reviewed case studies (e.g. [2, 3, 5-7, 10, 13, 14, 18-24, 26, 29-35, 37]), field studies (e.g. [4, 36]), and experience reports (e.g. [8, 15, 16, 25]).

4. Characterising the company contexts

We here give an overview of the different European company contexts used for our own case studies into successful DD. Our first set of case studies targeted GSD, in which temporal, geographic and socio-cultural distances were all high. We supplemented these with studies of five companies in which temporal and socio-cultural distances were low, and geographic distance the primary issue. There are few documented case studies of development which is distributed primarily geographically. Such is the case for "near-shoring" [8] and in particular for intra-national collaboration. In avoiding high temporal distance, and in particular if there is a low socio-cultural distance, a development context is significantly changed. Additional case studies were therefore conducted to explore such contexts, which would be more typical of an intra-European collaboration.

The first GSD company is primarily a hardware company, whose secondary software activities support their hardware, providing functionality for their customers. The company, which has over 90,000 employees worldwide, was established over 35 years ago, and established their presence in Ireland through the acquisition of a local company. Here, the software development teams work with other teams based at sites including the US, Malaysia, China, India, and Poland. In the past, certain projects included up to eight global sites. This experience of working with teams globally has given the team in Ireland much experience in GSD. The company has developed tactics over time for dealing with the challenges of GSD, which can provide great insight into effective software development in a distributed environment.

The second GSD company provides financial services and investment resources internationally and is one of the largest private companies in the US. The company has been developing software in Ireland since 2001, and currently employs around 200 people at its site in Galway. The software products developed are supplied to internal customers in the US, and involve coordinating with several software development teams in the US and others in India. There are efforts within the Irish team to implement follow-the-sun development through daily hand-offs of the software development tasks between sites.

The third GSD company can trace its origins back to the 1930s, providing desktop support services right through to mission critical service delivery. Specifically, the Irish team develop remote support and proactive services. Interestingly, the Irish site is the result of an acquisition, and later, a global merger. Therefore, the organisation is of special interest due to its transition through several different companies, allowing for an insight into problems due to organisational cultural differences. This company's approach to GSD can be more closely compared to global virtual teams, with one team effectively split across two sites in two continents.

The fourth company, which is part of a large company in the automotive sector, is an IT company with around 5000 employees world wide. The company develops different types of systems including business administration systems and manufacturing control systems. The site for the study had about 150 employees. The distributed team was in the process of rolling out a new development methodology (RUP) at the time; the method group was located at the main site.

The fifth company is part of a large telecom corporation. The company is divided over several sites both within Sweden and internationally. The largest and dominant site has about 1600 employees. The case study was of a project between two sites within Sweden.

The sixth company is a medium sized, geographically distributed IT company within a large enterprise focused on software intensive systems. It has approximately 230 employees and covers a broad spectrum of business areas such as defence, aviation, automotive, medical and telecoms. The company has a long experience of systematic method work and model based systems development.

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: University of Limerick. Downloaded on May 12, 2009 at 04:33 from IEEE Xplore. Restrictions apply.

The seventh company is a small company specialising in the development of biomedical information systems, and technical and business IT. They use MDA development technologies, with development split over two sites in the North and South of England.

The eighth company is a small IT company which primarily develops bespoke systems, in concert with other companies for large systems. They use open and inner source[1] development and utilise open source software frameworks including Castor, Webworks and Velocity.

5. A reference model for successful DD

In this section we present a reference model for successful DD. The model is evolved from the case- and field-studies in DD, informed by ten strategies for DD developed earlier. The reference model is in the form of a rich description of two exemplar companies. One aims to characterise an exemplar company in a GSD context, whereas the other aims to characterise one in an intra-continental DD scenario (in other words where the temporal distance is typically low, such as for nearshoring). Hereafter, we will refer to the former as GSD and the latter as intra-national DD. It should be noted that the reference models are expressed as an ideal type of which no actual organisation may exist. It can be used as a reference against which to compare existing DD practice in a company, or as a reference point for companies interested in moving into DD.

The GSD reference model is developed for a typical international GSD company. The intra-national DD reference model is developed for situations in which DD is adopted in a context in which temporal and socio-cultural distances are low and geographical distance dominates. Such a scenario would typify most intra-European collaborations. All case- and field-studies referred to above have been interpreted in developing these two exemplar models.

5.1 Concerning strategy 1 – Have a clear distribution rationale

Summary of the strategy: 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.

Motivation – *GSD only*: All distributed projects have been proceeded with on the basis of one or more of the following:

- there is a cost advantage to distributing
- the collaboration offers an opportunity to leverage time zone differences, for example for disaster recovery
- an existing team can be scaled cost-effectively, without dramatic impact on it
- the project gains access to skilled employees from top universities
- the company gains through closeness to more customers
- the company was able to keep key workers who had chosen to move

Motivation – *intra-national DD only*: All distributed projects have been proceeded with on the basis of all except the second bullet above (leveraging time zone differences).

Lowering risk – *both GSD & intra-national DD*: Risk of distribution has been lowered for each project by ensuring that the following are already in place:

- a mature team
- clear procedures and processes.

All projects have strict requirements and are large enough to modularise and amenable to splitting into appropriate units of work (not too large or small). Projects are not over-distributed: the smaller a unit the lower the gain in efficiency. Certain phases of the development lifecycle are used in preference to others (e.g. testing) as they are easier to distribute.

Selecting sites – *GSD only*: True cost is recognised as different from apparent costs. Groups claiming to be CMM level 5 may not be so in practice, so are checked. Some geographical areas are known to suffer from high attrition rates, so managers travel for interviewing key staff. The added costs of socio-cultural distance have been accounted for.

Planning distribution of teams – *GSD only*: Each site has a critical mass: an effective cluster containing inspirational people capable of motivating others. Teams have been constructed based on known expertise in sites. Teams are not split over more than two geographical zones. Ideally a focus has been given for each site, based on who is where. Lower value-added work is carried out where it is cheapest, but potential for peaky workload has been considered. A considered decision has been made on whether to keep sites relatively independent or deliberately encourage close collaboration. Consideration has been given to possibilities of remote delivery and management of

[1] This term is used to indicate the use of Open Source processes within a closed virtual organisation, see for example [12, 17].

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: University of Limerick. Downloaded on May 12, 2009 at 04:33 from IEEE Xplore. Restrictions apply.

customers' systems. For planning working time overlap it has been established whether the culture is one amenable to flexibility with respect to working hours. For company roll-out of a project deliverable, all sites using it have been involved. This may have been through an inner or open source initiative. In the case of open source, the company understands open source values and preconditions for utilising open source software, and has been able to form a symbiotic relationship with the community in participating in development.

Planning distribution of teams – *intra-national DD only*: Each site has a critical mass: an effective cluster containing inspirational people capable of motivating others. Teams have been constructed based on known expertise in sites. Ideally a focus has been given for each site, based on who is where. A considered decision has been made on whether to keep sites relatively independent or deliberately encourage close collaboration. Consideration has been given to possibilities of remote delivery and management of customers' systems. For company roll-out of a project deliverable, all sites using it have been involved. This may have been through an inner or open source initiative. In the case of open source software, the company understands open source values and preconditions for utilising open source software, and has been able to form a symbiotic relationship with the community in participating in development.

5.2. Concerning strategy 2 – Clarify all understandings

Summary of the strategy: 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 inter-organisational processes and stabilising processes.

Documentation – *both GSD & intra-national DD*: Documentation is kept very tight, particularly that associated with status reporting and communication plans. Documentation in a language that is non-native to producers is filtered through native speakers whenever possible. The role of documentation and ways to document at different sites are understood, agreed upon and maintained.

Clarity – *both GSD & intra-national DD*: Great clarity is maintained over roles and responsibilities, and also expectations. To aid in this, all decisions of meetings (including especially teleconferences) are documented clearly in minutes within a shared repository.

5.3. Concerning strategy 3 – Leverage modularity

Summary of the strategy: A system architecture mirrors the structure of the organisation which built it (Conway's law), so for software development work ensure that the architecture of the system is consistent with the distributed structure of the team. This may significantly impact on architectural decisions, but will reduce the need for intensive collaboration and allow optimum utilisation of local skills. For other life-cycle phases plan natural divisions of work in relatively small bundles.

A project is either partitioned functionally or by process within the development lifecycle used by the company.

Functional partitioning – *both GSD & intra-national DD*: Partitioning for discrete functional parts reduces feature dependency across sites, reducing the need for communication (once or twice a week typically).

Units of work are packaged to reduce the need for coordination, not too small yet not large enough to frustrate.

"Ownership" of largely independent work packets is handed over to a remote site, rather than breaking packages down further. Groups are structured correctly for the allocation strategy, e.g. consistent with its architecture.

Distribution by life-cycle phases – *both GSD & intra-national DD*: Work is focused on certain phases in the distribution life-cycle, e.g. QA-type testing. However, the site will have been engaged in the project as early as possible – in this example to design the test cases.

5.4. Concerning strategy 4 – Use cultural mediation

Summary of the strategy: Training in cultural issues is useful. Beyond that, use a cultural mediator. This is a person from one team context spending time in another, and becoming a link person between the teams. Many GSD teams use such '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.

Close to client – *GSD only*: Clients will have a single point of contact. Distribution may allow this to be made close by (geographically and culturally) if there is a local site.

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: University of Limerick. Downloaded on May 12, 2009 at 04:33 from IEEE Xplore. Restrictions apply.

Close to client – *intra-national DD only*: Geographical closeness may still be a significant issue. Customers can also show strong regional preferences.

Past enculturation – *GSD only*: Ensuring that there is an employee from the other region, or a team member with experience of working in the other region, helps with learning the culture.

Increase awareness – *GSD only*: Managers, particularly from the main site, are encouraged to visit other sites where they have responsibilities. On-site cultural training is undertaken. For example, managers will be made aware that in some cultures it is polite to agree with managers, but that this does not necessarily mean understanding or ability to deliver.

Increase awareness – *intra-national DD only*: Managers, particularly from the main site, are encouraged to visit other sites where they have responsibilities.

Develop cultural mediation – *both GSD & intra-national DD*: Periods of relocation both ways between sites is used to build cultural awareness, as well as to improve the sense of individuals being part of a team.

Language training – *GSD only*: Language can be the hardest problem in meetings. Language training is undertaken to reduce future communication problems.

Language training – *intra-national DD only*: Language can still be a problem in meetings. Whenever needed, language training is undertaken to reduce future communication problems, particularly if the company has an official language which is not the first language of its country.

5.5. Concerning strategy 5 – Facilitate human communication

Summary of the strategy: 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.

Synchronous – *both GSD & intra-national DD*: A context switch can cost time (some say 15 minutes), so protocols for synchronous interaction will be evolved. Instant messaging is used for quick questions but is not allowed to become intrusive (presence is not seen as a right to interrupt). Phone is used if the subject is a little vague, and also for monthly 1:1 meetings with peer-level managers or more regular "buddy" contact. Whilst on the phone, sharing of documents is often necessary; email, NetMeeting or similar will be routinely used. Teleconferencing is used for meetings (planning and technical). Regular teleconferences (typically twice weekly for projects) should be from your own desk. Video is currently seen as ineffective so rarely used. Informal chat before meetings is encouraged for ice-breaking. VoIP is only in beta and is not yet considered efficient enough for use, and Skype, a common VoIP application, presents security issues.

Asynchronous – *both GSD & intra-national DD*: Email is used for more complicated and more formal communication, but is used cautiously in order that it does not get out of hand. It is seen as allowing a time lag in response. Long email chains are problematic – in a distributed environment chains remain active after work finishes at a site. Using only emails for communication with a site or person is not practiced: it is not considered to be very productive. Centralised project information Project information is permanently available, and shows all national holidays across sites, to allow planning.

Centralised project information – *both GSD & intra-national DD*: Project information is permanently available, and shows all national holidays across sites, to allow planning.

5.6. Concerning strategy 6 – Manage processes

Summary of the strategy: 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 timeshifting. Synchronising 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.

Time shifting – *GSD only*: Follow the sun is not used for software development. Instead, employees often time shift to increase (or establish) the time window during the working day for synchronous communication. Meetings are scheduled to fit the agreed window (e.g. local meetings in the morning in Europe, meetings involving USA in the afternoon). Time zones are sometimes leveraged for handovers at the end of a working day, expecting results at the commencement of the next day. Time zone differences invariably lead to either late working with evening calls from home, or very early working. Out-of-hours

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: University of Limerick. Downloaded on May 12, 2009 at 04:33 from IEEE Xplore. Restrictions apply.

email checks are also common from home, to reduce potential delays otherwise incurred at the other site. This is supported by offering free Internet access and laptops to all employees.

Management – *GSD only*: Management meetings are held every week or two, and technical meetings every week – or more frequently during design reviews. There will also be 1:1 contact daily with team members. Anyone with a remote manager will have been assigned a local manager for HR issues. A more direct style is sometimes used when managing remote employees. It is recognised that virtual environments more complex to manage as people try to get away with more. Managers will also be aware of the fear and uncertainty within the original staff about exporting jobs.

Clear responsibilities will have been evolved, for example a single point of contact for a feature. Where possible, work schedules will be designed so that individuals are able to reschedule their own work in the event of a delay as the remote manager may not be available. This is to prevent delays in response bringing work to a standstill.

Management – *intra-national DD*: Most of the points made about GSD apply. Those that are deemphasised are the fear of exporting jobs, and concern about response delays bringing work to a standstill.

5.7. Concerning strategy 7 – Develop a sense of teamness

Summary of the strategy: Common strategies include the development of a project home page, which includes team member details and important planning information such as national holidays. Also summarise 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.

Team building *both GSD & intra-national DD*: It takes time to develop relationships, so resources are allocated to help preserve them, not least through effective communication of team activities and project progress. All members know who is in the team, which is brought together occasionally for training etc. – commencing with key individuals at the start. There is an awareness of the tendency to underestimate the ability of other sites (particularly offshore sites in the case of GSD), making them feel like sub-teams, and this is actively countered. Mechanisms are in place to encourage greater involvement, such as promoting specialist communities. Developers are paired across sites, in various ways: perhaps as buddies, pair programmers or with one in a mentoring role.

Project home page – *both GSD & intra-national DD*: The project home page maintains a list of roles and who is responsible for each feature in the project, and also personal and professional profiles, and photos.

5.8. Concerning strategy 8 – Encourage temporary collocation

Summary of the strategy: 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) synchronisation and review meetings, but front-loading travel is considered most effective. Variation includes project phasing, with one phase distributed and another phase in-house.

When – *both GSD & intra-national DD*: Members from a team are brought together early if a new partner site is involved. Other than this, certain phases are recognised as ideal: project initiation and completion; design phase; integration phases; training. In planning collocations, the long-term advantages of trips in both directions are recognised by the company – by developers as well as managers.

How long – *both GSD & intra-national DD*: Even short periods can be an advantage, but secondments of up to two years are seen as effective; from three to six months is more normal for training.

How often – *both GSD & intra-national DD*: Travel is restrained by cost, but trust tends to decrease in the absence of face to face meetings so it is supported if a manager requests it.

Complications – *GSD only*: Travel to the main site may be delayed through visa restrictions, so is planned well in advance. It is recognised that temporary collocation can sometimes result in permanent relocation, and not always within the company.

5.9. Concerning strategy 9 – Address heterogeneity

Summary of the strategy: 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

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: University of Limerick. Downloaded on May 12, 2009 at 04:33 from IEEE Xplore. Restrictions apply.

project-level confusion. One suggested strategy is to provide an interactive process model which can be tailored for each team.

Common tools – *both GSD & intra-national DD*: Enforcing common tools and processes makes collaboration much easier, but there is a recognised cost in retraining and the delay in attaining maximum productivity from sites currently using different tools and processes.

5.10. Concerning strategy 10 – Develop an effective tool base

Summary of the strategy: 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.

Code repositories – *both GSD & intra-national DD*: A single code repository is enforced, with local tailoring. It is mirrored to address the problem of cross-site backups.

Other repositories – *both GSD & intra-national DD*: Design artefacts and other documents are also maintained in a central, shared repository. This reduces delays in response because of visibility of current status.

Support tools – *both GSD & intra-national DD*: Appropriate collaborative development tools are provided, for example to support defect tracking and the gathering of quality metrics. There is standardisation on core technologies, such as visual change control. There is still a need to reduce the technologies used, but strong views are held.

Outsourcing problems of info sharing – *both GSD & intra-national DD*: Outsourcing brings with it the added problem of access control to the project repositories and so is usually avoided.

Remote access to systems – *both GSD & intra-national DD*: Collaboration is made more difficult when a remote developer needs access to systems: e.g. tunnelling through firewalls. Expertise is maintained to facilitate such situations.

6. Conclusions

In this paper we have considered eight case studies and the peer reviewed literature on DD processes, specifically focusing on case studies and field studies in DD. The intention has been to group and characterise the strategies proposed from real-world experience for reducing risk in DD and thereby leveraging its opportunities.

We have presented a reference model which can be used as a reference point for any company wishing to review their own distributed development scenario. This is particularised in two forms, one as an exemplar model for a GSD development scenario and one a particularisation of this for intra-national DD scenarios. It is apparent that, although there are a number of situations that fall between those (e.g. intra-continental), we have shown that an 'ideal' situation in a company is similarly characterised both in a GSD situation (where the temporal and geographical distances are typically high) and in an intra-national DD situation (where instead geographical and temporal distances are rather limited).

7. Acknowledgements

This research has been financially supported by the European Commission via FP6 Co-ordinated Action Project 004337 in priority IST-2002-2.3.2.3 'Calibre' (http://www.calibre.ie), and by the ITEA project COSI (Co-development using inner & Open source in Software Intensive products) (http://itea-cosi.org) through Vinnova (http://www.vinnova.se/), and also by the Science Foundation Ireland Principal Investigator projects B4-STEP and Lero.

8. References

[1] 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 (Paris, August 29, 2005). Austrian Computer Society, 2005, pp. 47-61.

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

[3] Aman, A. and Nicholson, B. "The process of offshore software development: preliminary studies of UK companies in Malaysia", In IFIP TC8 & TC9/WG8.2 & WG9.4 Working Conference on Information Systems Perspectives and Challenges in the Context of Globalization, Kluwer, Dordrecht, 2003, pp. 201-216.

[4] Bass, M. and Paulish, D. "Global Software Development Process Research at Siemens", In The 3rd International Workshop on Global Software Development, (co-located with ICSE 2004), pp. 11-14, <gsd2004.cs.uvic.ca/docs/proceedings.pdf >

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

[6] Boland, D. and Fitzgerald, B. "Transitioning from a Co-Located to a Globally-Distributed Software Development Team: A Case Study and Analog Devices Inc.", In The 3rd

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: University of Limerick. Downloaded on May 12, 2009 at 04:33 from IEEE Xplore. Restrictions apply.

International Workshop on Global Software Development, (co-located with ICSE 2004), pp. 4-7, <gsd2004.cs.uvic.ca/docs/proceedings.pdf>
[7] Borchers, G. "The Software Engineering Impacts of Cultural Factors on Multi-cultural Software Development Teams", In Proceedings 25th International Conference on Software Engineering, IEEE Computer Society, Los Alamitos, 2003, pp. 540-545.
[8] Carmel, E. and Agarwal, R. "Tactical approaches for alleviating distance in global software development", IEEE Software, 18, 2, 2001, pp. 22-29.
[9] Carmel, E and Tjia, P "Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce", Cambridge, NY: Cambridge University Press, 2005.
[10] Casey, V. and Richardson, I. "Practical Experience of Virtual Team Software Development", In European Software Process Improvement (EUROSPI) 2004, Trondheim, Norway.
[11] Clark, H.H. "Using Language", Cambridge University Press, Cambridge, 1996.
[12] "COSI Overview", Co-development using inner & Open source in Software Intensive products, ITEA-project, http://www.itea-cosi.org/
[13] 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, IEEE Computer Society, Los Alamitos, 2003, pp. 793-794.
[14] Damian, D.E. and Zowghi, D. "The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization", In Proceedings IEEE Joint International Conference on Requirements Engineering, IEEE Computer Society, Los Alamitos, 2002, pp. 319-328.
[15] Ebert, C. and DeNeve, P. "Surviving Global Software Development", IEEE Software, 18, 2, 2001, pp. 62-69.
[16] Espinosa, J. A., Cummings, J. N. and Wilson, J. M. "Research on Teams with Multiple Boundaries", In Proceedings of the 35th Hawaii International Conference on System Sciences – 2002, IEEE Computer Society, Los Alamitos, 10p.
[17] Fitzgerald, B. "The Transformation of Open Source Software", MIS Quartery, 30, 3, 2006, pp. 587-598.
[18] Ghosh, T., Yates, J.A. and Orlikowski, W.J. "Using Communication Norms for Coordination: Evidence from a Distributed Team", In Twenty-Fifth International Conference on Information Systems, Association for Information Systems, 2004, pp. 115-127.
[19] Heeks, R., Krishna, S., Nicholson, B. and Sahay, S. "Synching or Sinking: Global Software Outsourcing Relationships", IEEE Software, 18, 2, 2001, pp. 54-60.
[20] Herbsleb, J.D. and Grinter, R.E. "Splitting the Organization and Integrating the Code: Conway's Law Revisited", In Proceedings of the 21th International Conference on Software Engineering (ICSE'99), ACM Press, New York, 1999, pp. 85-95.
[21] Herbsleb, J.D. and Grinter, R.E. "Architectures, Coordination, and Distance: Conway's Law and Beyond", IEEE Software, 16, 5, 1999, pp. 63-70.
[22] Herbsleb, J. D., Mockus, A., Finholt, T. A. and Grinter, R. E. "Distance, Dependencies, and Delay in a Global Collaboration", In CSCW 2000 – ACM 2000 Conference on Computer Supported Cooperative Work, ACM Press, New York, 2000. pp. 319-328.
[23] Herbsleb, J.D., Mockus, A., Finholt, T.A. and Grinter, R.E. "An Empirical Study of Global Software Development: Distance and Speed", In Proceedings of the 23rd International Conference on Software Engineering – ICSE 2001, IEEE Computer Society, Los Alamitos, 2001, pp. 81-90.
[24] Imsland, V., Sahay, S. and Wartiainen, Y. "Key issues in Managing a Global Software Outsourcing relationship between a Norwegian and Russian firm: Some Practical Implications", In 26th Information Systems Research Seminar in Scandinavia, Finland, 2003.
[25] Karlsson, E.-A., Andersson, L.-G. and Leion, P. "Daily build and feature development in large distributed projects", In Proceedings of the 2000 International Conference on Software Engineering: ICSE 2000 the New Millennium, ACM Press, New York, 2000, pp. 649-658.
[26] Kiel, L. "Experiences in Distributed Development: A Case Study, In International Workshop on Global Software Development: GSD 2003", (co-located with ICSE 2003), pp. 44-47.
[27] Lings, B., Lundell, B., Ågerfalk, P.J. and Fitzgerald, B. "Ten Strategies for Successful Distributed Development", In The Transfer and Diffusion of Information Technology for Organizational Resilience, IFIP Volume 206, Springer, Boston, 2006, pp. 119-137.
[28] Lundell, B. Lundell, B., Lings, B., Ågerfalk, P. and Fitzgerald, B. "The Distributed Open Source Software Development Model: Observations on Communication, Coordination and Control", In The 14th European Conference on Information Systems, Gothenburg, 12-14 June, 2006 [CD-ROM].
[29] 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.
[30] Nicholson, B. and Sahay, S. "Some political and cultural issues in the globalisation of software development: case experience from Britain and India", Information and Organization, 11, 1, 2001, pp. 25-43.
[31] 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, IEEE Computer Society, Los Alamitos, 2005, pp. 1-10.
[32] Orlikowski, W. J. "Knowing in Practice: Enacting a Collective Capability in Distributed Organizing", Organization Science, 13, 3, 2002, pp. 249-273.
[33] Paasivaara, M. "Communication Needs, Practices and Supporting Structures in Global Inter-Organizational Software Development Projects", In International Workshop on Global Software Development, (co-located with ICSE 2003), pp. 59-63, <gsd2003.cs.uvic.ca/gsd2003proceedings.pdf>

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: University of Limerick. Downloaded on May 12, 2009 at 04:33 from IEEE Xplore. Restrictions apply.

[34] Paasivaara, M. and Lassenius, C. "Collaboration Practices in Global Inter-organizational Software Development Projects", Software Process Improvement and Practice, 8, 2003, pp. 183-199.
[35] Paasivara, M. and Lassenius, C. "Using Iterative and Incremental Processes in Global Software Development", In Third International Workshop on Global Software Development GSD 2004, (co-located with ICSE 2004), pp. 42-47, <gsd2004.cs.uvic.ca/docs/proceedings.pdf >.
[36] 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), pp. 69-74, <gsd2003.cs.uvic.ca/gsd2003proceedings.pdf>
[37] Vähäniitty, J. and Rautiainen, K. "Towards an Approach for Managing the Development Portfolio in Small Product-Oriented Software Companies", In Proceedings of the 38th Hawaii International Conference on System Sciences, IEEE Computer Society, Los Alamitos, 10pp, 2005.

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: University of Limerick. Downloaded on May 12, 2009 at 04:33 from IEEE Xplore. Restrictions apply.