Experiences in Global Software Development – A Framework-based Analysis of Distributed Product Development Projects

Lane, Michael T.  
*University of Limerick and Lero – The Irish Software Engineering Research Centre*  
Michael.lane@ul.ie

Ågerfalk, Pär J.  
*Uppsala University*  
*SWEDEN*  
par.agerfalk@dis.uu.se

Abstract

Many authors have reported on various challenges and benefits encountered by teams engaged in Global software development (GSD). Previous research has proposed a framework to structure these challenges and benefits within dimensions of distance and process. In this paper, the framework was used as an analytic device to investigate various projects performed by distributed teams in order to explore further the mechanisms used in industry both to overcome obstacles posed by distance and process challenges and also to exploit potential benefits enabled by global software development.

1. Introduction

Ågerfalk et al. [1] conducted an analysis of published GSD literature. The focus of their research was on three particular processes that have received a lot of attention in relation to distributed development: communication, coordination and control. They defined communication as being the transfer of information between parties in a manner that enables all parties to achieve a common understanding. Coordination addressed the management of dependencies between people and tasks. Control referred to project management issues such as task duration, quality, scope and cost.

References in the literature to the above processes were further investigated along the lines of three dimensions of distance: time, space and socio-culture.

Temporal distance addressed the overlap in working hours between two distributed parties. Geographical distance considered the accessibility of locations between collaborating parties – this not only included physical distance but also issues such as transport infrastructure or political border controls. Socio-cultural distance spoke to the perception of one party about the values and practices of a remote party. An individual’s organizational or national culture could impact upon their perceptions of the other party - a low socio-cultural distance between parties would indicate that they perceive the other as having similar values and norms – this would assist their communication.

Table 1 presents an overview of a framework produced from the above research [1]. This framework is used to analyse various experiences of the first author in GSD projects. It is a 9-box matrix that looks at the aforementioned areas of communication, coordination and control from the perspectives of geographical, temporal and cultural influences.

2. Background

2.1. Development teams

Software development teams may be categorized as either “Information Systems” (I.S.) teams or “Packaged” teams. The construction of internal or bespoke software systems tends to be the main activity of I.S. teams whereas Packaged teams build COTS products that are sold to wider markets. [3]

All experiences described in this paper reflect on distributed development conducted within Packaged teams. As such, it is likely that application of the GSD framework to I.S. teams may reveal further insights.

2.2. GSD Configurations

Distributed teams may be configured in different structures to promote best use of resources and to deal with the idiosyncrasies of the product under development [4]. A modular structure would permit teams to control development of a subsystem and establish clear interfaces, hopefully leading to a solid overall architecture. Functional expertise-based structures enable effective allocation of resources based upon different roles. Customization-based structures facilitate product-focused central development while distributed teams work on client sites. This structure aims to meet client needs while
avoiding bespoke product development that fails to meet wider market demand. Phase based structures would organize distributed teams to perform different phases of the systems development lifecycle. Time-zone based configurations can help to transfer work through the 24-hour day (“follow the sun”) and enable shorter overall task duration and consequently optimize project duration. The various experiences listed in Section 3 demonstrate the application of different structures to different product development situations.

### Table 1: Overview of the Framework for GSD issues [1]

<table>
<thead>
<tr>
<th>Process</th>
<th>Dimension</th>
<th>Temporal Distance</th>
<th>Geographical Distance</th>
<th>Socio-Cultural Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication</td>
<td>Reduced opportunities for synchronous communication, introducing delayed feedback. Improved record of communications.</td>
<td>Potential for closer proximity to market, and utilisation of remote skilled workforces. Increased cost and logistics of holding face to face meetings</td>
<td>Potential for stimulating innovation and sharing best practice, but also for misunderstandings.</td>
<td></td>
</tr>
<tr>
<td>Coordination</td>
<td>With appropriate division of work, coordination needs can be minimised. However, coordination costs typically increase with distance.</td>
<td>Increase in size and skills of labour pool can offer more flexible coordination planning. Reduced informal contact can lead to reduced trust and a lack of critical task awareness.</td>
<td>Potential for learning and access to richer skill set. Inconsistency in work practices can impinge on effective coordination, as can reduced cooperation through misunderstandings.</td>
<td></td>
</tr>
<tr>
<td>Control</td>
<td>Time zone effectiveness can be utilised for gaining efficient 24x7 working. Management of project artefacts may be subject to delays.</td>
<td>Difficult to convey vision and strategy. Communication channels often leave an audit trail, but can be threatened at key times.</td>
<td>Perceived threat from training low-cost ‘rivals’. Different perceptions of authority/hierarchy can undermine morale. Managers must adapt to local regulations.</td>
<td></td>
</tr>
</tbody>
</table>

2.3. Organization studied

All the cases reported in this study involved packaged teams from one particular organization and dealt with projects conducted between 1998 and 2005. The corporation had 24 offices worldwide including software development facilities in various US locations (California, Washington, Minnesota, Michigan), Ireland, Spain, France, Mexico, Brazil, Belgium, Netherlands, UK, China, India and Australia. Developed products had a worldwide presence and were used in customer facilities in over 84 countries. Motivations for global software development include access to customer feedback, use of cost-effective resources and the incorporation of remote resources due to acquisitions [2]. All of these motivations were evident in this organization as it sought to meet market needs for COTS products worldwide.

Certain characteristics of the packaged teams studied were consistent with the findings of [3]: there were ample resources available to the Packaged teams to ensure many different software development roles engaged in each of the projects. Also, a very strong awareness of project durations existed throughout the development organization. Product release dates were a major factor in all planning decisions. However, other characteristics of these teams conflicted with Carmel and Sawyer’s findings: Packaged teams adhered to strict software development guidelines and were required to conform to established development processes and standards. Each of the Packaged teams was responsible for the delivery of modules of COTS products. In certain cases, the teams were highly “projectized” – distributed team members tended to remain on standing teams and report directly to the project manager. In other cases, matrix structures existed and distributed team member reported directly to local management and indirectly to a central project manager (i.e. a “dotted-line” relationship). [5]

Carmel proposed a number of “centripetal forces” or solutions that help make GSD work. These included the aforementioned GSD configuration, collaborative technology and software development methods [2]. However, it should be noted that these forces might also provoke issues that could impact upon team performance. Problems of trust can emerge in certain team structures [4]. Partial resistance to adoption of collaborative technologies can reduce the efficiency of a group [7]. The advantage provided by clear
development methods in a GSD environment is that they help to form a common “language” for all parties. However, this should be balanced with the risk of introducing large learning curves as each remote site comes to terms with a new process. [8] Table 2 briefly describes how these forces were represented for each of the projects under analysis. The table also contains additional factors which may have influenced project performance: team size and level of customer involvement.

### Table 2: Projects studied

<table>
<thead>
<tr>
<th>Case</th>
<th>Team size</th>
<th>Team Structure</th>
<th>Customer Involvement Level</th>
<th>Development Environment</th>
<th>Software Development Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>80–100</td>
<td>Phase / Function</td>
<td>Minimal – key customers engaged as early adopters prior to general release.</td>
<td>Formal environment built internally using commercial configuration management systems</td>
<td>ISO approved corporate-wide internal process.</td>
</tr>
<tr>
<td>C</td>
<td>10–14</td>
<td>Module / Function</td>
<td>Minimal – mainly driven by internal product managers /business analysts</td>
<td>Formal environment built internally using commercial configuration management systems</td>
<td>Project based tailored version of corporate-wide internal process</td>
</tr>
<tr>
<td>E</td>
<td>40–50</td>
<td>Module</td>
<td>High customer involvement. Maintained live bespoke version of product in parallel with COTS product development.</td>
<td>Early version of commercial framework – inadequate documentation and environment instability impacted on productivity.</td>
<td>Specialized development process created to deal with environment – many updates to process due to framework instability and customer needs.</td>
</tr>
</tbody>
</table>

### 3. Experiences (cases in parentheses)

Particular experiences from the five projects outlined in Table 2 are presented in this section. All descriptions represent reflections of the first author on these projects, using table 1 as a lens to frame his thoughts. The first author is in a strong position to propose these views of the projects as he played a leading, guiding and reflective role (Lead designer/Development manager) in all of the projects.

#### 3.1 Cell 1 Communication × Temporal (A)

Distributed teams often suffered delays due to the unavailability of remote colleagues in different time zones. Further delays were incurred due to the misinterpretation of emails or voicemails. It became a regular practice to prevent potential delays by supporting asynchronous communications with telephone calls (often made from employee’s homes after working hours). However distance issues also motivated a discipline of creating effective documentation that could be easily shared with remote team members and reviewers. This documentation became a rationale history that proved useful for other purposes including technical training materials for maintenance staff and reference materials for the development of later product releases.

Time zone effectiveness was achieved in situations where tasks were very clearly defined and handovers were either not required or could be performed efficiently. An example of an activity suited to twenty-four hour global development was test execution. A daily build process mandated the execution of a build verification test. The purpose of this test suite was to test all programs within the various subsystems developed by distributed teams worldwide. The duration of the test script was sixteen hours. The script was structured to execute tests in a particular order so that modules were tested in the same time zone as the
teams responsible for them. Failures could be quickly resolved by these teams, thus allowing the script to be re-started from the failure point. If development had been restricted to an eight-hour day, script failures could have resulted in non-production of the daily version of the product.

3.2 Cell 2 Communication × Geographical (A)

The first author participated in many impromptu design sessions. These sessions were far more successfully performed by co-located design teams than distributed teams which is consistent with Carmel’s point about the challenges posed by distance [2]. Team telephone conferences that were allowed to delve into design issues often rapidly degenerated into confusing arguments. Remote design reviews appeared to benefit from the need for formality and clear preparation required by remote communication. But creativity sessions often floundered in their attempts to reach common understandings on various abstractions. These issues were evident despite access to collaborative technologies such as application sharing technologies and network-enabled “smartboards”. Common frames of reference (such as published patterns) provided by use of a commercial framework failed to alleviate the issue. In high-priority situations, the only recourse was to fly different designers to a common location, which usually resulted in rapid positive resolution to the design problems (but required additional costs).

Carmel’s assertion of the necessity of technological support to GSD [2] was confirmed in many situations. Corporate network facilities and groupware technologies were used effectively to manage reviews and processing of project artefacts. Instant messaging applications proved especially effective. They indicated the availability of remote colleagues, facilitated application sharing and allowed the recording of synchronous communications.

One particular distributed team used a functional distribution structure to establish a group of performance tuning experts responsible for supporting all other development. The performance group were located in close physical proximity to the laboratories of major vendors such as IBM, HP and Sun Microsystems. This structure ensured that the global development organization benefited from the relationships between their benchmarking team and the vendors of the main platforms used by their products.

3.3 Cell 3 Communication × Socio-cultural (B)

Other centrifugal forces quoted by [2] include poor depth of communication caused by context dilution and communication errors due to cultural differences. Evidence of this was seen in the case of a remote maintenance group whose productivity steadily deteriorated over time. Regular communication did not highlight the issue. An investigation uncovered inefficiencies due to work being performed on obsolete versions of products. This was attributed to a cultural tendency of diligent adherence to defined processes.

In another situation, reduction of cultural distance risk was achieved by the temporary co-location of remote team members at the project outset. Other aspects of this initiative were the establishment of a remote leader, mandatory daily meetings and intermittent social occasions. The remote leader and core development manager worked closely to prevent potential misunderstandings. Following the remote team’s return to their local office, mandatory daily communication was maintained but the teams also benefited from informal lateral communication [6].

3.4 Cell 4 Coordination × Temporal (C)

Reduced overlap of hours between locations was addressed by reservation of these times for remote communication. No local meetings could be scheduled during these times. This proved effective for many years when the bulk of communication was between groups in Ireland and the USA. However, the company culture was eroded as Irish developers began to regularly interact with teams from Europe, Africa, Australia and China. It became increasingly difficult to depend upon any core hours for either local or remote collaboration. Effective time management necessitated the use of scheduling software.

3.5 Cell 5 Coordination × Geographical (D)

Certain product development organizations structured their staff into module-based teams. These teams showed evidence of strong cohesion and pride in producing quality artefacts prior to their review by the wider development community. They also preserved the integrity of modules by disciplined use of clearly established interfaces in communications between their respective sub-systems. Some distributed teams were motivated to establish standards due to communication obstacles they faced in performing reviews.

Carmel described coordination breakdowns, loss of face-to-face communication and lack of “teamness” to be potential outcomes of GSD [2]. One technique used to prevent misinterpretations of critical tasks was crisp mandatory daily meetings. These meetings proved very effective in promoting informal communication of task dependencies and also had the side effect of nurturing a sense of “teamness”. A key outcome was that each team member had an up-to-date understanding of the
critical task path. This technique proved most effective with module based team configurations.

Carmel also described a common methodology as a binding force that can promote effective GSD [2]. One project empowered local teams to tailor their own processes while committing to supply certain standard artefacts of the global process. This promoted a common language for the entire distributed team while respecting and preserving the experiences that drove each local team in the selection of their own process.

### 3.6 Cell 6 Coordination × Socio-cultural (E)

Issues of trust and coordination were evident in a distributed team that used a customisation structure. The structure consisted of a core development team and customisation teams at customer sites. The core development group were suspicious of the ability of customisation groups to implement the product due to the steep learning curves required to both use it and configure its technology stack. An additional major concern was the instability of the underlying commercial framework and lack of supporting vendor documentation in relation to patterns and framework extensions. Placement of a core team member on the customisation team resolved this issue. It helped to overcome knowledge and documentation deficits and also build a connection into the development team for the remote development group.

### 3.7 Cell 7 Control × Temporal (E)

Scheduling of certain artefact review meetings was difficult due to the reduced availability of different distributed team members. Many project dependencies were seriously impacted when additional review meetings were required due to the failure of an artefact to pass its review. One day of rework could result in a two-week delay due to scheduling difficulties.

### 3.8 Cell 8 Control × Geographical (A)

A phase-based distributed structure proved unproductive for some phases and effective for others. Location of activities such as requirements, design and coding in different sites experienced delays due to the need for extensive inter-site communication. However, it proved very effective to establish a separate site for final product systems testing – due to the well-defined comprehensive artefact handover process.

### 3.9 Cell 9 Control × Socio-cultural (B)

A further reason for the deterioration in maintenance described in 3.3 was due to management’s method of gathering information. Management were accustomed to a horizontal open structure of information sharing and failed to recognize the hierarchical culture of the maintenance team.

### 4. Lessons learned

Certain lessons emerge from the cases described above. Different product development situations can be dealt with more effectively by application of appropriate distributed team configurations. However, care needs to be taken when selecting the configuration. Customization teams may be helpful in protecting core development while addressing client needs, but there needs to be an awareness that customization teams could be left isolated if core development is ongoing and complex. Section 3.1 describes how “Follow-the-sun” appears most useful when handovers are minimized.

As GSD organizations evolve, managers should monitor communications to ensure that practices that proved effective in earlier development efforts remain so. Additional geographical, temporal or cultural locations may demand that past practices be supplemented/improved to cater for new issues. Care needs to be taken to distinguish communication mechanisms used in creative processes versus processes used to review work.

### References