Experiences in Distributed Development: A Case Study

Lori Kiel
Interdisciplinary M.A. Candidate, Departments of Anthropology and Computing Science
University of Alberta, Edmonton, Canada
lkiel@ualberta.ca

Abstract

The findings from a case study involving a mid-sized software development organization illustrate the complex interaction of factors common to many global software development projects. The focus of the study is a single product development group that was distributed between two international divisions of the company, one in Canada and one in Germany, for a twenty-month period. When the distribution ultimately failed, it was a web of social, cultural, linguistic and political factors, rather than use or misuse of specific tools or techniques, that emerged as being most significant in the project's ultimate demise. A summary of these factors is presented here.

1. Introduction

Despite a recent decline in the international software industry, there is every reason to believe that there will be continuing pressures towards the adoption of globalized approaches to software creation. These approaches may take the form of formalized outsourcing agreements; they may emerge as collaboration among various divisions of international organizations; or they may consist of a small group of individual programmers who work together but live in separate cities.

Consequently, managers are frequently asked to organize software development projects that draw upon a mix of personnel in multiple locations. The technical barriers to such practices are diminishing rapidly. What about the human factors? Given the newness of the phenomenon, internationally distributed software development projects, whether they work, and why they either succeed or fail are little understood.

The study described here was undertaken in order to contribute to the small but growing body of empirical knowledge of global software development.

2. The Company

The subject company is a medium sized software development firm with offices in Canada, the United States, Germany and Malaysia. While some of the international offices were originally created within the corporate umbrella, others were acquired within the last two or three years. Consequently, the company has a diverse and dispersed set of resources, and managing that diversity represents a significant challenge.

Given this international structure, it is not surprising that the organization has undertaken distributed projects in one form or another for much of its history. Many of these projects have been short term, or structured on an *ad hoc* basis according to shifting delivery and development pressures. One project, however, represented a departure from this pattern.

For a period of roughly twenty months, spanning October 1999 to May 2001, the company structured all development of one of its core projects according to a fully distributed model. Before that time, they used multiple teams in their German and Canadian offices creating customized versions of this core application for each client. Teams in Canada developed for North American clients, while teams in Germany handled clients in Europe. In order to reduce duplication of efforts and facilitate a comprehensive code re-use strategy, they pooled their expertise in the two offices. A single distributed team was created; it was given the task of producing a new version of the product that would be the base of all future developments.

Ultimately, the distribution failed, and the company consolidated all development for the base product in a single office. While a base version was successfully created, the significant overhead required to support the distribution was deemed to be too expensive.

The purpose of my study was to answer the question, how did the distribution of software development for this product fail?

3. Method

In order to answer this question, I completed a qualitative case study in which interviews constituted the primary data collection technique. Senior managers in the company developed a comprehensive list of all project participants from which seventeen individuals were selected for participation in the study. These seventeen participants represented a cross-section of the development process, including managers, business analysts, trainers, team leads and developers. I interviewed all seventeen

individuals, recorded those interviews where possible, and used the transcripts and interview notes as the basis of analysis.

The strength of qualitative research is its ability to engage with multiple perspectives as experienced by participants in a particular set of circumstances, and to isolate recurring patterns or characteristics. To that end, I created codes that reflected the patterns that emerged from a preliminary review of the data. These codes were further refined upon a second review of the data.

In total, I applied twenty-three codes to the data. They fit into three main categories that form my analytical framework. The first consists of the main *themes* of distributed development that have emerged from the data. These themes represent the abstract or broad conceptual challenges to distribution that participants described.

The second category consists of the *practices* or specific ways of doing things on the ground, on a day-to-day basis within the company as described by the participants. It is at the level of practices where the themes are realized and reinforced.

Finally, a third category contains general observations made by the participants about the process of distributed development. This category contains the participants' own reflections on the overall practice of distribution, not necessarily their own experiences with a particular distributed project or projects.

4. Themes

I identified five main themes in the interview data: time, language, power, culture and trust. These themes define the broad, conceptual challenges to distributed software development that emerged in this case study. Taken individually, none of them is particularly surprising. The cumulative impact is, however, remarkable.

4.1. Time

Overwhelmingly, participants reported that the eight-hour separation in time zones between the two offices presented a substantial challenge to the project. When people were arriving at the office in Germany (8am), people in Canada (midnight) had long since left for home. The workday was just starting for the Canadian developers as things wound down in Germany. Depending on the schedules of particular individuals, it was possible for there to be little or no overlap in the workday.

The most often reported consequence of this temporal separation was a reliance on asynchronous communication techniques, primarily e-mail. Problems that should have been simple to resolve often dragged on for days. What might have been settled by a quick conversation was often blown out of proportion because the information needed in order to resolve the situation had to be communicated through e-mail, a message that might not be read for as much as 16 hours. Furthermore, the conversion of ideas and arguments into e-mail form introduced great opportunity for misunderstanding, particularly problematic when the content of the communication was contentious or argumentative.

4.2. Language

The main business language of the company is English. Not surprisingly, language emerged as a point of tension for almost all of the German participants, as well as for those participants from the Canadian office who had spent considerable time in Germany. The most common observation was that it was very difficult to fully participate in a teleconference in English. Often such meetings were oriented to some sort of problem solving or dispute resolution and, as such, could be dynamic and highly charged. Voices were raised, and people spoke rapidly. German participants reported frustration at not being able to follow or participate in the discussion. Canadians often interpreted the silence coming from the other office as an indication that no one in Germany wanted to participate or add to the discussion, and carried forward with the meeting. Many of the Canadian participants described being in meetings in Germany where they could see that the majority of the people participating in the teleconference were having great difficulty following the discussion.

In addition, many of the German participants reported a reluctance to engage in argument over the telephone. When technical or methodological debates arose – a necessary component of any software development activity – German speaking participants reported that they preferred to have the time to formulate their position, write it down, check it, ensure that they were saying what they meant to say and, finally, send it off in an e-mail. While this addressed their discomfort, it introduced the potential for misunderstanding and stretched out the problem-solving exercise over an extended serious of asynchronously exchanged e-mails.

4.3. Culture

Many participants – from both offices - suggested that people in the two offices could be characterized as exhibiting a particular set of attributes they identified with the term *culture*. They indicated that people perceive things differently, people say things differently, and people make things differently. Participants identified cultural differences as having posed a challenge to the project.

For example, the Germans were described as being blunt, efficient, stubborn, and to the point, but reluctant to speak out and criticize openly. On the contrary, the Canadians were described as being laid back, chatty, comfortable with open criticism, lax and indecisive. Several participants reported that the Canadians often

interpreted the Germans as being rude, and that the Germans were often frustrated by the Canadian way of doing things. Most individuals acknowledged the differences in national culture, and were aware of how their own behavior might be interpreted by people in the other office. Despite this awareness, frustration with the behavior of the other group persisted.

4.4. Power

One issue that was not necessarily explicitly stated, but emerged in some form in many conversations was the issue of power. Many participants reported that, particularly in the early days of the distribution of the project, decisions were made in the Canadian office and flowed to the German office. According to one participant, the Canadian office has historically seen itself as the 'brains of the operation'. Managers were typically located in the Canadian office. On many occasions, managers from the Canadian office temporarily relocated to Germany, but there were no managers from within the ranks of the German office.

The flow from Canada to Germany was not restricted to management personnel and management decisions: technical standards and architectural decisions usually originated in the Canadian office and the developers in Germany were expected to adopt these standards and accept the decisions. Many participants reported a strong resistance on the part of the German developers. These developers often refused to follow standards or use tools developed in other offices. One senior manager reported with frustration on the outright intransigence on the part of some in the German office. Another offered that not following standards was almost a point of pride, and suggested that managers failed to recognize the importance of local ownership of standards. The history of the company is characterized by a marked imbalance of power between the Canadian and German offices.

4.5. Trust

Temporal separation, language gaps, cultural differences and inequality between the two offices all contributed to the challenge of building an atmosphere of trust, respect and cooperation that characterizes a cohesive software development team. Common to participants from both offices were stories of misunderstandings, angry exchanges, and complete dismissal of people in the other office as incompetent. Indeed, such tensions seem to extend well beyond the specific boundaries of the project of concern for my case study. Many participants reported general problems in dealing with the other office, regardless of the particular project. One participant described the friction between the two offices.

Several participants suggested that when you are physically separated from co-workers, it is easy to ignore them and devalue their contributions and abilities. At great distance it is difficult to empathize with those in the other office, and this in turn makes it challenging to maintain an atmosphere of mutual respect and shared understanding. Several people commented that it often seemed that simple choices – for example, the timing of a meeting - completely failed to consider the impact on those in other offices.

Participants generally acknowledged that the people in the other office were not, in fact, incompetent: everyone recognized the abilities of their co-workers. Nonetheless, several people noted that when denied access to the context in which a decision was made and the detailed reasoning that entered into the decision, it became very easy to dismiss an apparently poor choice as being the work of an idiot. This was particularly the case in a crunch situation when everyone was under pressure to meet a deadline.

5. Practices

Along with the five themes identified above, I have isolated several practices that articulated with the themes, dramatically impacting the experience of distributed development for the project participants. Many of these practices involve specific tools such as e-mail, telephone, teleconference, StarTeam (configuration management tool), net-meeting, translation tools, and the company intranet. Others refer more to business processes or ways of doing things, like software process, project team size, project scope, management practices and travel for face-to-face meetings. There is a complex interaction among the various themes and practices that must be explored if anything is to be learned about this particular distributed software development exercise.

For example, while the practice of using e-mail has several basic characteristics that exist in any circumstance, certain properties become critical when it is employed in a predominantly asynchronous and politically charged communication environment. Other qualities emerge when messages are exchanged between people from different cultural or linguistic backgrounds. And still other factors are important when messages are exchanged between close friends. Examination of these contextual elements is fundamental to understanding how e-mail and other practices shape and are shaped by the distributed software development project.

Teleconferencing is another practice that deserves careful consideration. Anyone who has participated in a teleconference can identify with the awkwardness of this form of dialogue. It's often difficult to control speaking order, people frequently talk over one another, and equipment problems can make people in the other location difficult to understand. The data in this study indicate that when linguistic and cultural differences are present, along with inter-office power struggles, these problems are dramatically amplified.

Managers at the organization recognized the challenges presented by the project's linguistic, cultural and political

context, and tried many things to mitigate the emerging problems. For example, English language training was made available in the German office, with some success. In addition, developers were repeatedly encouraged – "prodded" – to phone their colleagues when problems needed resolving, rather then send them messages via e-mail. Unfortunately, the time zone separation limited the effectiveness of this approach.

The project team leaders employed a simple but powerful tool to create links among team members: the company intranet. They created a site where they posted photographs of everyone working on the project. Their goal was to decrease the personal distance between the teams. It was generally well received, and many participants commented on the positive impact that this had on the team.

Another practice that many respondents had positive comments about was the opportunity to travel to the other office. Such travel provided an occasion to get to know and work with individuals from that office. They made high-bandwidth, face-to-face meetings possible, and provided an excellent opportunity for what one participant called cross-pollination between the offices: an opportunity to exchange ideas about how the thing that everyone is building should be built. When people spent time with one another, the cultural, and linguistic barriers began to break down, leading to less conflict. According to one senior level manager, these exchanges were 'like gold'. Many people reported that whenever team members spent time in the other office it was a successful team-building occasion.

Despite these efforts, it was team-building that emerged as one of the biggest challenges for this project. Numerous respondents indicated that, while they put many processes into place to coordinate and control day-to-day activities, and while these processes were successful for the most part, process was not enough. Throughout the twenty-month term of the distributed development experiment, a strong sense of a single team never emerged.

6. Participants' General Comments

Most participants eagerly voiced their general opinion about their experience with distributed software development. They talked about communication in a very broad sense, incorporating language, time-zones, culture, as well as one or more practices. It was generally recognized that the success of any programming activity depends on successful communication and doing global work requires a certain overhead to maintain communication channels. According to one participant, when times are tight economically, it is these maintenance activities that are easiest to cut. Sooner or later, however, there are problems in taking this approach. Issues will pop up down the road.

Several participants used the term *bandwidth* to describe the character of the communication channel between various teams and offices. A high bandwidth link exists between individuals located in the same city, or between cities separated by a minimal number of time zones, whereas the link between the Canadian office and the German office has a very low bandwidth. Bandwidth is further degraded by cultural or linguistic distance.

Acknowledging the extra overhead inherent with distribution, several people remarked that it was simply too expensive for a single team, working from different offices, to develop a single product.

7. Conclusion

It is always compelling to try and isolate the one factor - the silver bullet [1] – that will solve a problem. In this case, the problem is how to use globally distributed technical personnel to create software in an effective and economical manner. For this study, no single factor can be isolated as the cause of the failure. For example, while temporal separation on its own might not present an insurmountable barrier to successful distribution, the cumulative impact of an eight-hour time zone difference, a subsequent reliance on asynchronous communication, and a poor inter-office relationship appears to have been a significant obstacle. Similarly, while teleconference meetings are admittedly a challenge for most of us, they nonetheless function as a powerful and effective communication tool for many international organizations. This effectiveness is severely diminished when language or cultural barriers hinder the degree to which all people involved in the meeting can participate fully.

In future work, I will analyze this data from an anthropological perspective, using the anthropology of technology as a general framework from which to consider the technology of distributed software development. This analysis will employ a decidedly expanded view of technology that considers not only the servers, telephones, teleconferencing equipment, code repositories, management practices and software processes that structured the daily activities of the software developers, but the social and cultural context in which these tools and processes were employed.

In the absence of this analysis, it is my hope that this summary of study findings will make a useful contribution to discussions of global software development.

8. References

[1] Brooks, Frederick P, *The Mythical Man-Month*, Addison Wesley, Berkeley, 1995.