An Empirical Study of Global Software Development: Distance and Speed

James D. Herbsleb, Audris Mockus
Bell Labs
Lucent Technologies
263 Shuman Boulevard
Naperville, IL 60566 USA
+1 630 713 1869, +1 630 713 4070
{jherbsleb, audris}@lucent.com

Thomas A. Finholt
School of Information
University of Michigan
3218 Computing Center Building
Ann Arbor, MI USA 48109
+1 734 764 6131
finholt@umich.edu

Rebecca E. Grinter
Xerox PARC
3333 Coyote Hill Road
Palo Alto, CA 94304
+1 650 812 4818
grinter@parc.xerox.com

ABSTRACT

Global software development is rapidly becoming the norm for technology companies. Previous qualitative research suggests that multi-site development may increase development cycle time. We use both survey data and data from the source code change management system to model the extent of delay in a multi-site software development organization, and explore several possible mechanisms for this delay. We also measure differences in same-site and cross-site communication patterns, and analyze the relationship of these variables to delay. Our results show that compared to same-site work, cross-site work takes much longer, and requires more people for work of equal size and complexity. We also report a strong relationship between delay in cross-site work and the degree to which remote colleagues are perceived to help out when workloads are heavy. We discuss implications of our findings for collaboration technology for distributed software development.

Keywords

Global collaboration, software development, delay, speed, awareness, informal communication

1 INTRODUCTION

Communication and coordination issues in large software engineering projects have always been formidable (e.g., [5, 8]). Increasingly, engineers and managers must add the challenges of coordinating work across sites, spanning national, language, and cultural barriers (see, e.g., [6]). Driven by market and resource requirements, the push toward globalization has generated a wide variety of problems for software developers [19].

Previous research [15, 17], suggests that cross-site communication and coordination issues cause a substantial loss of development speed. In this paper, we investigate relationships among delay, communication, coordination, and geographic distribution of work, in order to shed light on the possible mechanisms responsible for introducing delay.

A major software development effort at Lucent Technologies has been distributed for several years. It began with two sites in two countries in one continent. It has now grown to six primary sites located in four countries, in two continents, with an additional seventh supporting site in a third continent. This paper reports how problems with delays impacted this organization, and how the understanding achieved from this empirical research informed the development of tools to overcome these problems.

In the remainder of this introduction, we review literature on distributed work and how it differs from co-located work. In the following section we briefly describe our empirical methods. Then, we present new results on communication patterns across and within sites, and results showing the relationship of cross-site work, delay, and other important variables. Finally, we draw out the implications of these observations for achieving success in cross-site work, and conclude the paper.

1.1 Communication and distance

In sharp contrast to the popular image of software developers as relatively introverted and secluded, they in fact spend a large proportion of their time communicating. For example, in an empirical study of time use by developers in a large software engineering organization, Perry, Staudenmayer, and Votta [29] reported that "one of the most salient impressions conveyed by observation was the sheer amount of time each developer spent in informal

communication" (p. 41). The developers in their study spent an average of 75 minutes each day in "unplanned interpersonal interaction."

In an 8-month study of a medium-sized telecommunications software project [16], an analysis of time sheets indicated that about 50% of time was spent in "group work" (meetings and unplanned work-related discussions) during the first month, and this level dropped fairly steadily until only about 10% of time was spent in group work during the last month. Design activities, in particular, seemed to require a very large proportion of collaborative work (over 50% in all but one 4-week period), in contrast to the relatively solitary activities of coding and testing.

In contrast to the frequent interaction of co-located work, there is very convincing evidence that the frequency of communication generally drops off sharply with physical separation among co-workers' offices, and that the sphere of frequent communication is surprisingly small. Tom Allen [3], in a study of engineering organizations, reported that the frequency of communication among engineers decreased with distance. Further, he noted that when engineers' offices were about 30 meters or more apart, the frequency of communication dropped to nearly the same low level as people with offices separated by many miles.

Kraut et al [20] found similar results for scientists. Further, they found that the rate at which scientists collaborated spontaneously with one another was also a function of distance between offices, and that this effect was more powerful than the effect of same-discipline scientists tending to collaborate more frequently with one another. Presumably, the more frequent communications led to conversations in which common interests were discovered and acted upon.

These findings are particularly troubling in rapidly evolving, high technology environments, where the competitors, products, standards, and customers routinely create a demand for significant, unforeseen changes in requirements throughout the development cycle. In organizations with rapidly changing environments and "unstable" projects, informal communication is particularly important [12, 21]. For example, as requirements change, it is hard for the formal mechanisms of communication, such as specification documents to react quickly enough. Often news of change, its significance, and its potential impact, is propagated informally and very quickly among the development staff. Under such conditions, the pattern of lateral communication across sites should be particularly important in the development organization under study.

Research showing the importance of informal communication has lead to a variety of technologies designed to stimulate casual conversation among workers at different sites. These technologies have included video [1, 10, 11, 26], audio [18], and text [7]. To this list we must now also add instant messaging, a technology that has spread very rapidly, and is beginning to infiltrate the work place (e.g., [25, 30]). We have seen little indication, however, that these technologies have been widely deployed in software engineering organizations.

These observations about communication and distance also highlight the importance of understanding the dependencies among the various kinds of work involved in software development [13]. In a study of six software engineering organizations, Grinter, Herbsleb, and Perry [14] observed four different ways of organizing work across sites that evolved within a single global corporation. Each represented an attempt to minimize requirements for cross-site communication in the context of particular types of product architectures and mechanisms for coordinating work. There are also indications, from a study of an automotive engineering group, that, where possible, engineers will try to reduce the coupling of cross-site work [27].

In a case study of a software engineering organization spread across several sites, Herbsleb and Grinter [15] investigated how the organization used a number of mechanisms, including plans, processes, and interface specifications, to coordinate the cross-site work. Each mechanism, however, was vulnerable to imperfect foresight and unexpected events, which required substantial communication to coordinate activities and renegotiate commitments. Despite the need for communication, there was a nearly total absence of informal, unplanned communication across sites.

The difficulties of knowing who to contact about what, of initiating contact, and of communicating effectively across sites, led to a number of serious coordination problems. Among these problems were unrecognized conflicts among the assumptions made at different sites and incorrect interpretation of communications. The most frequent consequence of cross-site problems was *delay* in the resolution of work issues. By delay, we mean the additional time it takes to resolve an issue when more than one site is involved. So, for example, if a part of the design or code needs to be changed, or if someone needs a better understanding of how some part of the product works, people at more than one site may need to be involved in information exchange, negotiation, and so on, in order to find a solution. Such issues arise very frequently in software development.

Delays in resolving work issues can slow development considerably. Issues that would typically be resolved in hours or minutes often stretched out to days or weeks in the effort to find, establish contact, and have the necessary collaborative sessions with the right people to achieve resolution.

Qualitative studies (e.g., [15]), have shown how individuals are disrupted by cross-site coordination

challenges. But questions remain about the cumulative effects, for example, how distance affects the speed with which software engineering tasks are accomplished, and how distance is related to other important variables that influence speed, such as the size of a task, or the number of people involved. In addition to being important research questions, these are critical pragmatic issues as businesses become more globally distributed. Speed to market has become the most critical factor for succeeding with new products (e.g., [9]).

In this paper, we use two independent sources of data to examine the effect of distributed work on speed, and then examine a number of properties of cross-site versus same-site communication that may account for these differences. Finally, we discuss the implications of these findings for tools to address these communication issues.

1.2 Research questions

This paper reports a study of one geographically distributed organization, with particular attention to the effects of geographic distribution on delay in the development life cycle. We also examine the patterns and quality of communication in order to shed light on possible causes of delay.

Does cross-site work introduce delay, as compared to same-site work? Previous research suggests that working across sites introduces substantial delay because of reduced communication, difficulty in finding the right person and establishing contact, as well as having an effective collaborative session. We examine quantitative data comparing the time required for similar same-site and cross-site work.

What factors influence the time interval required to make a software change? What role, if any, does spreading work across multiple sites play in lengthening this interval? Assuming that there is an association between multi-site work and longer intervals, there are many distinct ways in which working across multiple sites might introduce this delay. By modeling the time interval required to make a software change, we extract evidence helpful in determining the nature of the relationship, and what causal mechanisms are plausible.

What differences are there between same-site and cross-site social "networks" and their effectiveness? Are they stable over time? One of the possible causes of multi-site delay is the difficulty of communication and coordination inherent in cross-site work. In order to begin to understand this issue, we address several basic questions about communication within a site and how it differs from communication across sites. For example, what is the relative size of local and cross-site social networks? How much instability is there in these social networks once they are established? Is there a perception of greater misunderstanding of tasks, priorities, plans, and changes across sites?

2 SITE AND METHODS

In this section we describe the sites of study, including some background on the products built. We also discuss how the work is divided among sites. We conclude with a description of the methods used to analyze and collect the data.

2.1 Sites

Geographically distributed software development is pervasive among most large technology companies, including Lucent Technologies. We chose one department of the company to study for three reasons:

First, the department was willing to host researchers and provide us with access to developers, documents, and source code. Second, they work in a complex area of telephony, where the market requirements and standards are changing rapidly. This makes coordinating the development work extremely difficult and subject to continuous change. In addition, this product competes in an aggressive market and that brings extreme time pressures to development work. Third, the department has cross-site development, described below.

In this study we focus on four locations, one in the UK, one in Germany, and two in India, where the department does a large share of its development work. These sites exchange information frequently and make decisions that require cross-site synchronization. The German site had existed for a number of years, and the people there had considerable experience working together on similar systems. However, it had not previously participated in cross-site development where parts of the product are split across sites. The UK site had only existed for about three years, and thus had no existing relationships to any other Lucent site. One Indian site was also about three years old. The other was a software contractor, not actually a part of Lucent, but it had worked with the German and UK sites for several years. With the exception of having only restricted access to the Lucent intranet, the contractor site participated fully in projects, in ways indistinguishable from Lucent sites.

The department also has interactions with other divisions of the company because the product must interact with other technologies. Many of these technologies are built in the United States so the developers coordinate work with these other sites. These US sites had not previously worked together, nor had they worked before with the UK or German sites. In all cases, the collaborations span different languages, cultures, and many time zones, making them more difficult.

2.2 Methods

Our results draw on modification request and survey data.

2.2.1 Modification Requests.

Like many software development organizations, the department we studied used a Change Management (CM)

system to organize and track its development work. CM systems organize development by providing mechanisms to ensure that developers coordinate changes they make to the software. Typically they provide mechanisms for versioning the code, and some ability to manage two or more developers making changes to the same software at the same time, in a structured way. CM systems track development work through correlating the actual changes in the code with requests to make those alterations. By following requests it is possible to see what changes were made to which parts of the software, whether all the changes were actually made, and who made them. It is because of the organizational and tracking features of CM systems that they present such unique opportunities to study collaborative work (see [13, 31]).

In the CM system that this development organization uses, the basic tracking unit is called the Modification Request (MR) which is a request to incorporate a specific functionality into the software. Some MRs ask for new functionality, others ask for specific problems or bugs to be fixed. All development work in the organization was done within the framework of an MR, using Sablime™ and ClearCase™. Moreover, processes surrounding the CM system were structured to support MRs.

The software used for tracking MRs automatically collects several valuable types of data. It establishes a record for each MR of who made the request, the date the request was made (or "opened"), and each change ("delta") that is made to the code base in order to fulfill the request. For each change, it records the login of the person submitting the code, and the time, size, and date of the submission. Large, complex changes typically have many deltas, whereas small, simpler changes have only a few, or even just one.

MRs are the basic unit of work in this software development. Moreover, MRs and their equivalents in other CM systems are pervasive in most software development work. By performing straightforward calculations on the MR data, it is possible to derive several important measures (e.g., [24]), such as the following:

Work interval. The difference between the date of the first delta and the last delta for an MR is a good approximation of the period of time, or interval, that was required to do the work of implementing the change.

Full interval. The difference between the date the MR was opened and the date of the last delta is a somewhat longer interval. It includes the work interval and also the time to determine whether to actually implement the change, to assign a priority, to assign the work to particular individuals, and for these individuals to actually start the work.

2.2.2 Survey

In November 1998, 117 employees located in Germany and the UK were invited to complete a Web-based questionnaire. Most of the workers were software engineers, with some managers and some administrative support personnel. In June 1999, a second administration of a similar survey was undertaken. In all, 160 employees in Germany, UK, and two sites in India were invited to take this survey.

The first questionnaire consisted of 68 items, the second had 65. Both included questions covering demographics, patterns of communication, working relationships, coordination, information exchange, and language. The respondents provided two answers for most questions: one with regard to local co-workers and the other with regard to distant co-workers. Many identical questions were included in both administrations of the survey. There were some deletions and additions, however, in order to drop questions that did not seem useful, to measure new variables, and to refine our measurements of others.

The surveys were administered in English in the UK and India. A German language version was produced using back translation techniques, and was available for German speakers. Both versions were pilot tested with members of the organization being studied.

Site	Survey 1, 1998	Survey 2, 1999
UK site	33	23
German site	41	39
India Internal	N/A	9
India Contractor	N/A	21

Table 1. Number of survey respondents by location.

Overall, 98 of 117 surveyed employees completed the first questionnaire, for a response rate of 83%[1]. Across the four sites, 160 employees were invited to participate in the second wave survey. We obtained usable responses from 96 individuals, for a response rate of 60%.[2]

3   RESULTS
3.1 Delay

We have two different measures of delay that allow us to compare single-site work with cross-site work and to validate different measures against each other. One measure is derived from our second survey, which included the following two questions:

> How many times in the past month was your own work delayed because you needed information, discussion, a decision, from someone at your site or another site?

---

[1] In this first survey, 22 of the responses were from sites we have not yet been able to visit. Because we were not certain we understood the relation of these sites to the two primary sites, these responses were eliminated from the 1998 survey.

[2] Four of the respondents reported no contact with any other site, so their data were eliminated.

What was the average length of the delays you experienced before acquiring the needed information, having the discussion, or being informed of the decision by the person from your site or the other site?

For each question, the respondent answered by supplying one number for "local site" and another number for "distant site." Of the 92 respondents, 39 reported at least 1 delay in the past month for the local site, and 48 reported at least one delay for the remote site. Averaged over all 92 respondents, the mean number of local delays was 2.1 delays per month, and the mean duration was .9 days. For cross-site delays, the mean number was 1.9 delays per month, and the mean duration was 2.4 days.

In order to test the significance of the differences in number and duration of local and remote delay, a paired observation t-test was performed on a square root transformation of the data[3]. The difference between the number of delays (local versus remote) was not significant (t=0.1758, df=91, not significant). The difference, however, in duration (local delays versus remote delays) was statistically significant (t=2.5079, df=91, p< 0.02). In summary, while there is no significant difference in the number of delays reported, their duration does vary significantly with delays crossing sites taking almost a day and a half longer than single site cases.

We see similar findings in the MR data. We extracted all of the single-site MRs, i.e., where everyone involved in the MR (the person who made the request and all the people who carried out the work of making the change) resided at one site, and compared them with the MRs which involved at least two sites.

The average single-site MR took about 5 days to complete, from the time the work actually began until the last change was made (work interval). In contrast, MRs which involved more than one site took 12.7 days, more than 2.5 times as long, to complete. If we look instead at the "full interval," i.e., the days it took to complete the request measured from the day the request was made, the difference between single-site interval (20.5 days) and distributed interval (27.1 days) is similar. (The full interval includes not only the time it takes to do the work, but also to review the request, assign it a priority, and assign the work.) The differences in interval are statistically significant (p < 0.001) using a t-test.

3.2 Modeling MR Interval

To understand potential mechanisms by which multi-site work might introduce delays, we used statistical modeling techniques to build a model of the MR interval.

---

[3] The scale for the delay data is truncated at zero, so the distribution is skewed, and consequently not suitable for a t-test. A square root transformation on interval produced a good approximation to the normal distribution and was used in the tests.

We selected a number of change measures that could be related to the change interval (see [24]).

- *Number of people*. We expect that the change interval may increase with number of people involved (a person opening an MR or making a delta) in the change because of potential communication and coordination issues.

- *Diffusion*. We expect that diffuse changes spanning large parts of the system would take longer to complete than localized changes, so we chose the number of modules touched by the change as the measure of diffusion.

- *Size*. We expect that larger changes are more likely to take more time to complete, so we chose number of delta to represent the size of the change.

- *Time*. We selected the time of the first delta as a time covariate to control for any time-related factors.

- *Bug fix*. We expect that bug fixes might have a different interval than other types of MRs.

- *Severity*. We included an indicator of high severity of an MR as a measure of priority (because the priority was not recorded). We expect that high severity MRs would also have high priority and, consequently, might be resolved faster.

- *Multi-site*. Finally, we expect that multi-site changes (involving people from more than one site) would take longer than single-site changes.

Due to extremely skewed distributions we performed a natural log transformation of interval, size, diffusion, and the number of people. The results of linear regression for work interval[4] are presented in Table 1.

Name	Predictor	Fitted	p-value
	Intercept	-7.7	< 0.001
# of people	Log(#people)	0.35	< 0.001
Diffusion	Log(#modules)	0.43	< 0.001
Size	Log(#delta)	0.15	< 0.001
Time	Date (years)	0.26	< 0.001
Severity	Is high severity	-0.12	< 0.01
Fix	Is fix	-0.08	< 0.1
Multi-site	Is multi-site	*	Not sig.

Table 1. Regression of log work interval; R=0.6, N=2227

The coefficients indicate that number of people, size, and diffusion significantly increase interval. The interval also increases with time and decreases with severity. Surprisingly, *given all other factors*, multi-site MRs do not have significantly longer intervals than single-site MRs.

There are several possible explanations for this unexpected finding:

---

[4] We performed a similar regression for the "full" interval and obtained essentially identical results, with the exception that severity had a much more significant influence on full interval.

1. *Large changes take longer to implement and also are more likely to involve multiple sites.*
2. *Changes touching many modules take longer to implement and also are more likely to involve multiple sites.*
3. *Multi-site changes require participation of a larger number of people, which, in turn, introduces delay.*

To investigate these hypotheses and to illustrate relationships among variables in our model we used graphical modeling techniques (see, e.g., [22, 32]). Figure 1 shows the result of stepwise fitting of graphical Gaussian (otherwise known as covariance selection) model with threshold p-value of 0.0001. The skewed variables were transformed to make their distribution more closely approximate the Gaussian distribution. The resulting model contains only links that have high values of deviance indicating significant partial correlations. The significance should be interpreted cautiously, of course, as in any technique involving model selection.

The nodes in Figure 1 represent variables, and links represent significant partial correlations among them. A partial correlation between variables X_1 and X_2 differs from a conventional sample correlation in that partial correlations show the strength of relationship between two variables *given* values of all other variables X_i that are directly connected to X_1 or X_2 in the model (see definition, in, e.g., [32]). The black color indicates positive partial correlations, while gray color indicates negative partial correlations. The thickness of a link shows its significance. The two numbers next to a link show deletion deviance (the higher the deviance – the higher the significance of the partial correlation) and partial correlation. The variables that are not directly related in the graph are independent given values of the other variables they have links to in the model. We refer to such variables as not directly related.

Figure 1 shows that the variables directly related to MR interval, from most to least significant, are number of people, diffusion, and size. It also shows that compared to single-site MRs, multi-site MRs tend to be associated with more people, and tend to increase in number over time (and are slightly more likely to be associated with bug fixes rather than new features).

Two of the plausible indirect relationships between the multi-site character of work and MR interval would appear to be ruled out by this model. In particular, the hypothesis that *large changes take longer to implement and also are more likely to involve multiple sites* is not supported by the analysis. Similarly, the hypothesis that *changes touching many modules take longer to implement and also are more likely to involve multiple sites* receives no support. The multi-site variable does not have a significant partial correlation with either size or diffusion.

The model does, however, support the third hypothesis that *multi-site changes require participation of a larger number of people, which, in turn, introduces delay.* Multi-site changes are strongly related to the number of people who work on an MR, a variable that, in turn, is strongly related to interval. It appears that splitting work across sites slows the work down primarily because it requires the involvement of more people than comparable work accomplished all at one site. To make sure that this trend is not simply caused by the fact that multi-site MRs *must* involve at least two people (and single-site MRs occasionally involve only one person), we fitted a graphical model excluding single-person MR's. The resulting model still indicated number of people as the most significant partial correlation for interval, and multiple sites as the most significant partial correlation for the number of people.
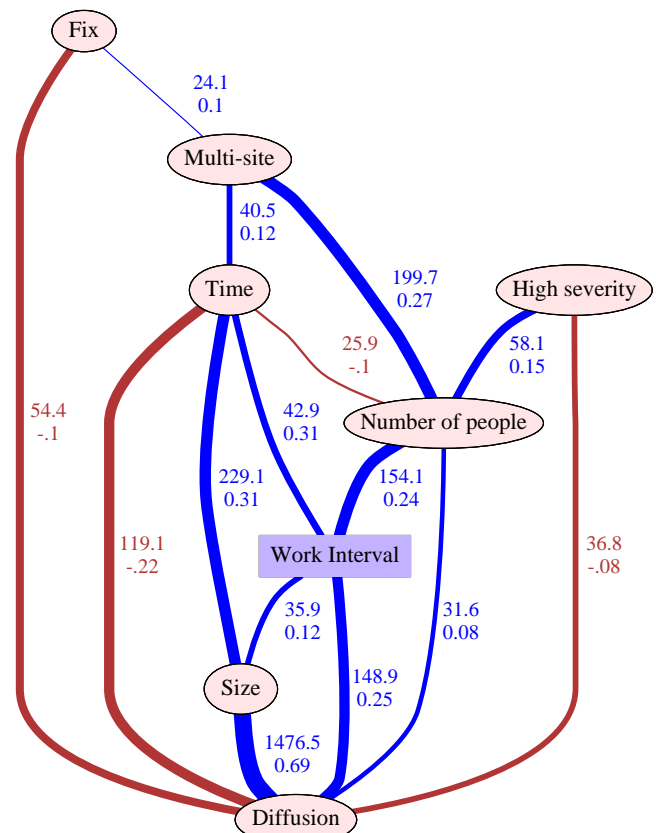

Figure 1. Graphical model of work interval[1].

The data reported in this section suggest that cross-site work takes substantially longer than same-site work, but if there is in fact a causal relationship, it appears to be mediated by the number of people required to do the work. Given these findings, it seems highly likely that characteristics of social "networks," and cross-site communication and coordination issues are critical to achieving speed in multi-site development. Among other questions, it is important to understand why multi-site

development appears to require more people than single-site development to do work of equivalent size and diffusion. In the following section, we focus on survey data relevant to these issues.

3.3 Communication

Size of personal networks. In order to get a rough estimate of the size of local and remote social networks, we asked people to

Consider an average week. How many different people do you typically interact with at work during the course of the week from your <local, remote> site?[5] (t=12.4036, df=77, p<.0001)

As we expected, the results were quite different for local and remote sites. The mean for local was 16.0, and for remote 4.9. This difference is highly significant.

How personal networks change over time. The two surveys included a "name generator" item that requested respondents to name up to 10 people he/she communicates with locally and up to 10 he/she communicates with remotely. We calculated the ratio of people each person named in both surveys to the total number of distinct people named by the person. The average ratio was 18% of the remote network and 22% of the local network, indicating a large turnover within a period of only seven months.

This figure is somewhat inflated for the local networks. The reason is that we only provided spaces on the survey for 10 remote and 10 distant colleagues. This is probably sufficient for remote networks (self-reported mean size of 4.9), but insufficient for local networks (self-reported mean size of 16). For local colleagues, the changes partly represent a difference in the samples of the total network that happened to be reported on the two occasions.

Effectiveness of communication. We asked questions in order to test previous observations about differences in informal communication occurring within a site and between sites. We asked several questions about the difficulty of finding and contacting people. For all of these questions, the differences were highly statistically significant, in the expected direction, favoring local communication. (Questions labeled S1 appeared on the first survey, S2 appeared on the second; items in boldface are used in a later analysis.)

(S1) I lose time trying to figure out who to contact regarding my work. (t=4.44, df=66, p<.0001).

(S1, S2) **People I need to communicate with are difficult to find.** (S1: t=2.82, df=67, p=.006; S2: t=2.59, df=68, p=.01).

There was a significant difference in responses for remote and local sites, with a much greater tendency to believe that people at remote sites were more difficult to find and to contact.

We also asked several questions designed to assess the extent to which important information flowed through the communication network, and about inadequacies and barriers.

(S1) I often get useful work-related information through casual conversations (t=5.44, df=64, p<.0001).

(S1) There have been times when I was accidentally excluded from information which was shared by my co-workers (t=3.56, df=61, p=.0007).

(S2) **My co-workers provide timely information about changes in current plans** (t=6.3, df=91, p<.0001).

Again, the differences in responses across sites were highly significant, and in the direction suggesting better communication within a single site.

We also asked several questions in order to determine what kinds of misunderstandings may have arisen locally and across sites. Interestingly, there was little evidence for misunderstanding at the level of specific tasks. Questions probing these issues revealed that workers were no more likely to disagree about task priorities or doubt the clarity of task assignments for remote workers that for local workers:

(S2) There is disagreement about task priorities. (t=1.09, df=91, not significant).

(S2) When work is assigned, everyone is clear about his or her task. (t=1.62, df=91, not significant).

A question at the level of overall plans, however, showed a perception of significantly greater clarity in plans for local collaborations:

(S2) **Plans for future action are clearly formed** (t=2.9, df=91, p=.005).

A related issue is the response to load balancing problems, and the willingness to help a co-worker when they become overloaded. We asked about the respondents' willingness to help out when a co-worker is experiencing a particularly heavy workload:

(S2) I assist my co-workers with heavy workloads, beyond what I am required to do.

The results show only a small, nonsignificant difference in answers between local and remote (t=1.05, df=91, not significant). However, we also asked:

(S2) **My co-workers assist me with heavy workloads, beyond what they are required to do.** (t=6.26, df=91, p<.0001)

For this question, the respondents reported a sizeable, highly significant difference between the help offered by their (more helpful) local and (less helpful) remote colleagues. The contrast in these two answers is quite revealing. While people generally seem to believe they assist local and remote colleagues equally, they report that

---

[5] Again, because these data were bounded at zero, we performed the t-test on a square root transformation.

they are much more likely to receive assistance from local than remote colleagues.

Communication and delay. In order to determine which of these differences (i.e., responses to the questions in boldface) are the most likely candidates for causing the much longer cross-site delays, we performed two linear regressions. We regressed all six variables (responses to the questions in boldface) where significant local versus cross-site differences were found. In the first regression, we regressed local communication variables on the square root of the length of delay for local communication. We performed an analogous regression for cross-site communication and delay variables. (We only used questions from the second survey since delay data were not collected on the first.)

In the case of local communications none of the predictors was significant. Table 2 shows the results for cross-site communication and delay. The only predictor among these cross-site variables is the response to whether co-workers assist me with heavy workloads. The higher the reported assistance, the lower the delay.

Cross-site Data	Value	Std	t-val	Pr(>|t|)
(Intercept)	2.2	0.7	2.9	0.005
Coworkers assist	**-0.3**	**0.1**	**-2.6**	**0.01**
Plans clearly	-0.04	0.1	-.5	*
Hard to find	-0.01	0.1	-0.09	*
Size personal	-0.02	0.02	-1	*
Get timely	0.09	0.1	0.8	*

Table 2. Regression of cross-site survey variables on duration of remote-site delay (* indicates not significant).

4 CONCLUSIONS

Speed is perhaps the single most important success factor in modern high technology businesses. In the face of the growing globalization of all facets of work, from virtual teams to virtual corporations, the demand for speed must be accomplished, if it is to be accomplished at all, in a geographically distributed environment.

It is important to keep in mind that these results might differ for organizations that distribute work across sites in different ways (e.g., [14]), or that use other business arrangements, such as outsourcing. The possible effects of such differences are open questions for future research.

4.1 Research questions

Does cross-site work introduce delay, as compared to same-site work? The answer would appear to be "yes, but indirectly." Both the survey data and the MR data indicate that work that spans sites takes longer than work that does not cross sites. The fact that both sources of data indicate substantial cross-site delays increases our confidence in this result. But the nature of this relationship is not necessarily simple.

What factors influence the time interval required to make a software change? What role, if any, does

spreading work across multiple sites play in lengthening interval? A graphical model showed that size, diffusion, and number of people were all directly related to MR interval. It does not appear to be the case that cross-site changes take longer simply because they are larger, or touch more modules. Rather, multi-site changes tend to involve more people changing the code than do comparable single-site changes, and number of people is in turn strongly related to length of interval. This suggest that multi-site communication, coordination, and/or social networks may differ from their single-site counterparts in a way that requires more people to participate, thereby introducing delay.

What differences are there between same-site and cross-site communication networks and their effectiveness? Are they stable over time? We identified several differences, including the size of the communication network, the difficulty of finding people, the reduced likelihood of getting timely information, the clarity of plans, and the reduced likelihood of receiving help with heavy workloads. Both remote and local social networks were extremely volatile, with about 80% change in seven months.

Interestingly, and perhaps ironically, the only factor statistically related to the length of delays (as measured by the surveys) was the reported lower levels of receiving help with especially heavy workloads. The irony is that this is one factor where we have data indicating that no one seems to perceive him/herself as part of the problem. In other words, people believe they are no less helpful to remote colleagues, but the same population of respondents reports they are less likely to receive help from remote, as compared to local, colleagues.

There are several possible explanations of this effect. One simple explanation is that local networks are larger, hence one receives more help locally. The fact that answers to the parallel question about help rendered by the respondent are *not* different for local and remote colleagues casts doubt on whether their responses are influenced by network size.

Alternatively, it may be that people attempt to be equally helpful, but help offered across sites is relatively ineffective, so the perception of the recipient is that little help has been rendered. Or it may be that it is difficult to accurately convey the urgency of a situation across sites, so the potential helper believes he/she is offering an appropriate level of help, but underestimates the problem. These possibilities have somewhat different implications for collaboration tools, since the first would require tools that help with effectively carrying out the work, while the latter places more importance on communication tools that help convey more context, and perhaps more emotional content. In any case, this is a particularly urgent problem since it is directly related to delay, and the responses

suggest that people in general will see no need to take any action.

4.2 Multiple sites, number of people, and delay

In general, the survey results suggest that communication and coordination are substantially disrupted across sites. There are several possible ways in which this reduced capacity for effective communication could result in the need for more people to accomplish work that could be carried out by fewer co-located people:

Introducing conflicts. It is much easier to design and implement software changes in such a way as not to conflict with other parts of the product when all relevant parts are local. As a result of countless informal interactions, developers are much more aware of the details of pieces of products built locally, and are much more likely to communicate with those who work on them. This greater awareness of the context in which they are working allows developers to avoid creating conflicts with existing code, or with other changes.

Informal participation in changes. It is relatively easy to consult with local developers who have responsibility and expertise for code related to the code being changed. It may be that after such informal consultation, the developers primarily responsible for the change also make the necessary changes in the related parts of the code. This informal, and presumably faster, means of implementing changes may not be possible across sites, where the necessary communication and trust are absent, and a more formal process must be followed, involving more people submitting code for the change.

Getting right people involved. It is much more difficult to know precisely who has what responsibilities and expertise across sites. It is also much more difficult to communicate precisely what the relationship is between a proposed change made to code at one site and the impact of that change on code at another site. Given these constraints, it is much less likely, when changes at one site necessitate related changes at another site, that the correct person is involved from the beginning. Compared to the single-site case, it is more likely that the first person will make some changes as requested, only later realizing that slightly different expertise is required. If it is more difficult to get the right people involved from the beginning, changes will presumably involve more people and also take longer.

We leave it to future research to distinguish among these and other possibilities. Regardless of the precise mechanism, tools and practices that increase the ability to find experts, communicate clearly, and to increase informal communication, should reduce these problems.

5   TOOLS FOR GLOBAL COLLABORATION

There are a number of hints in the data about the kinds of technologies that might be effective. The research reported in this paper is part of a larger, multi-year project that is developing tools in all of these areas.

Finding experts. In general, workers have difficulty finding the right people across sites (e.g., [2]). Unfortunately, the extreme volatility of the communication networks we observed in this study suggests that people will typically have very limited exposure to a relatively small number of remote people (compared to local social networks). Across sites, there is relatively sparse interaction, and correspondingly few opportunities to learn who does what, and who has what expertise, and to be aware of where they are now. There is relatively little opportunity to learn in depth about the expertise of those at other sites.

We are currently deploying a tool called Experience Browser which provides a visualization of the CM system, designed to make it easy to discover who has experience working on which parts of the code, and to get contact information for that person. (See also [23].)

Awareness. Instant messaging applications can give some indication of whether a person is available, i.e., logged on to a server, such as AOL Instant Messenger. It also provides lightweight chat capabilities, which makes it very easy to start up informal, text-based conversation. Generally, there is strong and increasing interest in such systems (tens of millions of users according to news reports [30]). We are currently deploying a version of an IM tool called Rear View Mirror that includes functionality for work groups [4]. It incorporates MUD-like group chat facilities, in order to provide an ongoing, continuous cross-site discussion forum, and to avoid the intrusiveness of instant messaging, which demands an immediate response from a single recipient.

Shared calendars can also play a central role by letting people at other sites know where someone is, when they might be free, and even who they have been meeting with [28]. We have deployed a web-based calendar tool (which we call CalendarBot) in the development organization, and it is receiving heavy use, i.e., thousands of hits per month.

We are currently trialing a prototype web-based tool that shows international clocks, online calendars, and contact information for groups of people the user wants to stay in touch with. It will also be linked with the development organization web site that is being organized around teams. Navigating to a team's home page will automatically allow the tool to display the appropriate clocks, calendars, and person information for that team.

Richer interaction. Finally, the finding about non-help during heavy workload and delay may argue for tools supporting richer interaction, i.e., high quality audio and video, that may be more effective in conveying the nuances of expression and emotion that allow more accurate determination of urgency.

7 REFERENCES

1. Abel, M. J., Experiences in an Exploratory Distributed Organization, in *Intellectual Teamwork: Social Foundations of Cooperative Work*, J. Galegher, R.E. Kraut, and C. Egido, Editors. 1990, Lawrence Erlbaum Associates: Hillsdale, New Jersey. 489-510.
2. Ackerman, M. S. and C. Halverson. Considering an Organization's Memory. in *Computer Supported Collaborative Work*. 1998. Seattle, WA: ACM Press 39-48.
3. Allen, T. J., Managing the Flow of Technology. 1977, Cambridge, MA: MIT Press.
4. Boyer, D. G., *et al.* Virtual Community Presence Awareness. in *CSCW '98 Workshop on Designing Virtual Communities for Work*. 1999. Seattle, WA
5. Brooks Jr., F. P. The Mythical Man-Month. *Datamation* 20, 12, 1974, 44-52.
6. Carmel, E., Global Software Teams. 1999, Upper Saddle River, NJ: Prentice-Hall.
7. Churchill, E. F. and S. Bly. It's all in the words: Supporting work activities with lightweight tools. in *GROUP '99*. 1999. Phoenix, AZ 40-49.
8. Curtis, B., H. Krasner, and N. Iscoe. A Field Study of the Software Design Process for Large Systems. *Communications of the ACM* 31, 11, 1988, 1268-1287.
9. Cusumano, M. A. and D. B. Yoffie, Competing on Internet Time. 1998, New York, NY: The Free Press.
10. Dourish, P. and S. Bly. Portholes: Supporting Awareness in a Distributed Work Group. in *ACM CHI'92 Conference on Human Factors in Computing Systems*. 1992. Monterey, CA.: ACM Press 541-547.
11. Fish, R. S., *et al.* Evaluating Video as a Technology for Informal Communication. in *CHI'92*. 1992. Monterey, CA: ACM 37-48.
12. Galbraith, J., Organizational Design. 1977, Reading, MA: Addison-Wesley.
13. Grinter, R. E. Recomposition: Putting It All Back Together Again. in *ACM Conference on Computer Supported Cooperative Work (CSCW '98)*. 1998. Seattle, Washington: ACM Press
14. Grinter, R. E., J. D. Herbsleb, and D. E. Perry. The Geography of Coordination: Dealing with Distance in R&D Work. in *GROUP '99*. 1999. Phoenix, AZ
15. Herbsleb, J. D. and R. E. Grinter. Splitting the Organization and Integrating the Code: Conway's Law Revisited. in *21st International Conference on Software Engineering (ICSE 99)*. 1999. Los Angeles, CA: ACM Press 85-95.
16. Herbsleb, J. D., *et al.* Object-oriented analysis and design in software project teams. *Human-Computer Interaction* 10, , 1995, 249-292.
17. Herbsleb, J. D., *et al.* Distance, Dependencies, and Delay in a Global Collaboration. in *CSCW 2000*. 2000. Philadelphia, PA
18. Hindus, D., *et al.* Thunderwire: A field study of an audio-only media space. in *Computer Supported Cooperative Work*. 1996. Boston, MA: ACM Press 238-247.
19. Special Issue, IEEE Software, Global Software Development, March, 2001.
20. Kraut, R. E., C. Egido, and J. Galegher, Patterns of Contact and Communication in Scientific Research Collaboration, in *Intellectual Teamwork: Social and Technological Foundations of Cooperative Work*, J. Galegher, R.E. Kraut, and C. Egido, Editors. 1990, Lawrence Erlbaum Associates: Hillsdale, NJ. 149-171.
21. Kraut, R. E. and L. A. Streeter. Coordination in Software Development. *Communications of the ACM* 38, 3, 1995, 69-81.
22. Lauritzen, S. L., Graphical Models. 1996, New York.: Oxford University Press.
23. McDonald, D. W. and M. S. Ackerman. Expertise Recommender: A Flexible Recommendation System and Architecture. in *ACM Conference on Computer Supported Cooperative Work*. 2000. Philadelphia, PA: ACM Press 231-240.
24. Mockus, A. and D. M. Weiss. Predicting risk of software changes. *Bell Labs Technical Journal* 5, 2, 2000, 169-180.
25. Nardi, B. A., S. Whittaker, and E. Bradner. Interaction and Outeraction: Instant Messaging in Action. in *ACM Conference on Computer Supported Cooperative Work*. 2000. Philadelphia, PA: ACM Press 79-88.
26. Obata, A. and K. Sasaki. OfficeWalker: A virtual visiting system based on proxemics. in *Computer Supported Cooperative Work*. 1998. Seattle, WA 1-10.
27. Olson, J. S. and S. Teasley. Groupware in the Wild: Lessons Learned from a Year of Virtual Collocation. in *Computer Supported Cooperative Work*. 1996. Cambridge, MA: New York, N.Y.: ACM Press 419-427.
28. Palen, L. Social, individual, and technological issues for groupware calendar systems. in *CHI '99*. 1999. Pittsburgh, PA
29. Perry, D. E., N. A. Staudenmayer, and L. G. Votta. People, Organizations, and Process Improvement. *IEEE Software* 11, 4, 1994, 36-45.
30. Slatalla, M., The office meeting that never ends. Plenary New York City, September 23.
31. Tellioglu, H. and I. Wagner. Negotiating Boundaries: Configuration Management in Software Development Teams. *Computer Supported Cooperative Work: The Journal of Collaborative Computing* 6, 4, 1997, 251-274.
32. Whittaker, J., Graphical Models in Applied Multivariate Statistics. 1990, Chichester: Wiley.