Communication and Organization: An Empirical Study of Discussion in Inspection Meetings

Carolyn B. Seaman, *Member*, *IEEE*, and Victor R. Basili, *Fellow*, *IEEE*

Abstract—This paper describes an empirical study that addresses the issue of communication among members of a software development organization. In particular, data was collected concerning code inspections in one software development project. The question of interest is whether or not organizational structure (the network of relationships between developers) has an effect on the amount of effort expended on communication between developers. The independent variables in this study are various attributes of the organizational structure in which the inspection participants work. The dependent variables are measures of the communication effort expended in various parts of the code inspection process, focusing on the inspection meeting. Both quantitative and qualitative methods were used, including participant observation, structured interviews, generation of hypotheses from field notes, statistical tests of relationships, and interpretation of results with qualitative anecdotes. The study results show that past and present working relationships between inspection participants affect the amount of meeting time spent in different types of discussion, thus affecting the overall inspection meeting length. Reporting relationships and physical proximity also have an effect. The contribution of the study is a set of well-supported hypotheses for further investigation.

Index Terms—Communication, defects, empirical study, inspections, organizational structure, process, productivity.

1 Introduction

Many factors that impact the success of software development projects still defy our efforts to control, predict, manipulate, or even identify them. One factor that has been identified [5] but is still not well understood is how information flows between developers. It is clear that information flow affects productivity (because developers spend time communicating) as well as quality (because developers need information from each other in order to carry out their tasks effectively) [16]. It is also clear that efficient information flow is affected by the relationship between development processes and the organizational structure in which they are executed. A process requires that certain types of information be shared between developers and other process participants, thus making information processing demands on the development organization. The organizational structure, then, can either facilitate or hinder the efficient (i.e., timely) flow of that information. These relationships among general concepts are pictured in Fig. 1.

The study described in this paper addresses the productivity aspects of communication. In particular, it empirically studies how process communication effort (the effort associated with the communication required by a development process) is influenced by the organizational structure (the network of relationships between developers) of the development project. In this paper, we examine the organizational structure of one particular project, the code inspection process used, and the time and effort associated with inspection meetings. The inspection process was chosen as the context for this study because it provided a good opportunity to observe technical communication between developers (i.e. during the inspection meetings). However, the study and evaluation of inspections is not the focus of this work.

Our findings indicate that organizational attributes are significantly related to the amount of time inspection participants spend in different types of discussions. The aim of this study is not to test or validate hypotheses about relationships between these variables, but to explore what relationships might exist and try to explain those relationships. Our contribution, then, is a set of *proposed* hypotheses, along with an argument, in the form of supporting evidence, for their further examination.

Although the importance of efficient communication, and its relationship to organizational structure, is well supported in the organization theory literature [8], [15], it has not been adequately addressed for software development organizations. Communication has been identified as an important factor in how developers spend their time [16], and some organizational characteristics that affect its efficiency have been suggested [5], [12]. Some, but surprisingly little, of the "process" work in software engineering has dealt with information flow or organizational structure [3], [4], [17]. It has been postulated that informal communication (e.g. water-cooler conversation) is usually more valuable in general than formal, interpersonal communication (e.g., inspection meetings) [13]. However, there is still a need for focused studies of formal communication (like this one) because, unlike informal communication, it can be planned for and controlled, if we know the factors that can be manipulated to make it more

• *C.B. Seaman and V.R. Basili are with the Institute for Advanced Computer Studies, Computer Science Department, University of Maryland, College Park, MD 20742. E-mail: {cseaman, basili}@cs.umd.edu.*

Manuscript received 3 July 1997; revised 25 Nov. 1997.
Recommended for acceptance by S. Lawrence Pfleeger.
For information on obtaining reprints of this article, please send e-mail to: tse@computer.org, and reference IEEECS Log Number 105336.

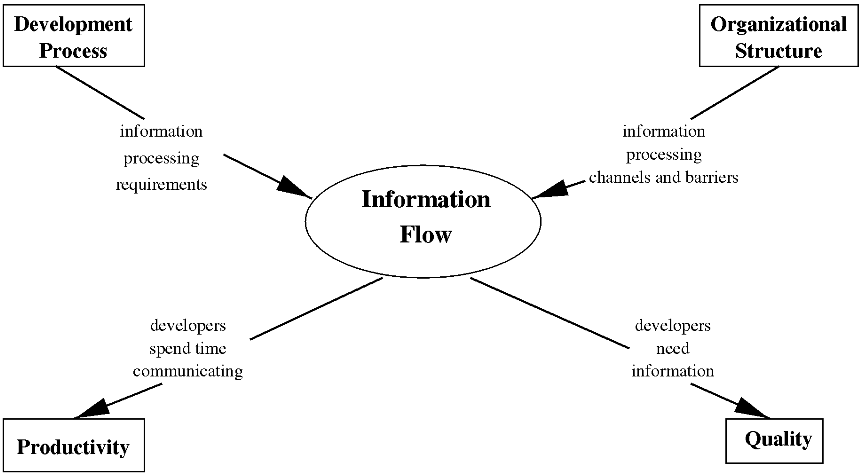


Fig. 1. Relationships between concepts relevant to this work.

efficient. Such studies must, by definition, be empirical studies because they deal with nonanalytical entities (i.e., people) that have few universally applicable laws or theories governing their behavior. Concepts such as communication, process, and organization must be studied where they occur, in real software development projects.

The study described here combines quantitative and qualitative research methods. Qualitative methods are designed to make sense of data represented as words and pictures, not numbers [10]. Qualitative methods are especially useful when no well-grounded theories or hypotheses have previously been put forth in an area of study. They are used mainly to generate, not test, hypotheses. Quantitative methods are generally targeted towards numerical results, and are often used to confirm or test previously formulated hypotheses. They can be used in exploratory studies such as this one, but only where well-defined quantitative variables are being studied. We have combined these paradigms to explore an area with little previous work, as well as to present hypotheses which are grounded on, or supported by, convincing empirical evidence. Because these research methods are not common in software engineering studies, we have chosen to describe them in some detail in Section 3.

2 Study Setting

The software project that served as the setting for this study was the development of a mission planning tool for NASA/Goddard Space Flight Center's Flight Dynamics Division (FDD). Much of the development was contracted to Computer Sciences Corporation (CSC). About 20 technical leads and developers (the most from CSC) participated in the inspection process during the course of the study, although more participated in the project. The project began in early 1995, the first release was scheduled for the summer of 1996, and it was delivered about a year late.

The two aspects of the development project of interest to this study are the inspection process used (in particular the communication required by that process), and the organizational relationships between the process participants. These are described below, along with the independent and dependent variables, which are closely tied to the organizational structure and the process, respectively.

2.1 Inspection Process

This study focuses on the development project's code inspection process. We relied initially on a written document, the *Code Inspection Procedure*, which defined the tailored inspection process for this project, including the relevant steps and roles. Throughout the study, however, we updated our understanding of the inspection process through observation and interviews.

Inspections were conducted after unit test, before submitting the code to configuration control, in preparation for integration testing. Both code (several C++ classes) and unit test products (test plan, test cases, test drivers, and results) were inspected. Some of the code inspected was produced by a code generator that was being developed concurrently with the mission planning application and thus could not be relied upon to produce error-free code. For this reason, the generated code was also inspected. Problems found during inspection with code generator output were reported to those responsible for implementing the generator.

The inspection meeting (the unit of analysis for this study) was one step in the inspection process. Inspection meeting participants included the "author," who had implemented and unit tested the C++ classes being inspected, the "moderator," a "code inspector," and a "test inspector." In some cases, more developers were assigned to inspect the code or test.

All the observed inspections occurred at CSC, and involved mostly CSC personnel. Inspectors were all developers chosen from the same development project by the moderator of the inspection. Most moderators reported that their choice of inspectors was based principally on availability, secondarily on perceived skill at inspection. Inspectors varied in terms of their experience (more on the possible effects of experience later). Inspections were not new to this group of developers, as nearly all of them had participated in inspections on previous projects. However, some of the administrative details of the company's standard inspection process were modified for this project, which caused some confusion early in the project.

The stated objective of the inspection meeting was to record defects in the source code (i.e., faults) that had been found by the inspectors during their preparation. This is similar in style to the inspection process proposed by Gilb and Graham in [9], except that discussion was not as severely restricted. On the contrary, there was a considerable amount of discussion during inspection meetings, mainly focused on precisely defining the defects, determining whether or not each issue raised was actually a defect, and discussing various technical issues related to the system being developed. The moderator of each inspection intervened when the conversation got severely off-topic, or when solutions were being discussed in detail, but otherwise there was little attempt to curb discussion. The process studied could not be considered a "Fagan-style" inspection process [7], in that there was no overview, and inspectors were expected to find defects during their preparation.

Although this study was not intended to evaluate the inspection process itself, it may be useful to understand something about the effectiveness of this process. This understanding can help us interpret the results of the study in light of the different ways in which people communicate when conducting an inspection. As explained later, neither operational defect data nor testing data are available for the code under inspection, so we cannot evaluate the number or percentage of defects missed during inspection. However, some indication of the inspection's efficiency can be provided by calculating the checking rate (number of pages reviewed per hour during the preparation period). Gilb and Graham observed empirically that the optimum checking rate for detecting defects is usually about one page per hour, although each organization should calculate their own optimal rate based on historical data. The average checking rate over the inspections in this study was around 60 pages per hour. This would seem to indicate an unusually ineffective inspection process, but the situation may not be as dire as the checking rate implies.

1) First, Gilb and Graham's figure of one page per hour is an optimum, and may have little relationship to checking rates normally achieved in practice. They point out, in fact, how difficult it is for many organizations to achieve this optimal rate. A checking rate of 60 pages per hour is actually closer to checking rates on some real projects reported in the literature (e.g., [2]).
2) Second, the pages being inspected in this study included a high percentage of white space and comments, which probably did not require as much time to inspect as executable statements.
3) Third, much of the code inspected was generated by a code generator, which created a great deal of code that was similar from one class to another (as well as much of the white space and comments). The generated code was inspected to ensure that the generator was working properly, which is a bit different from the usual purpose of inspection, so it's not clear that Gilb and Graham's optimal checking rate applies.

In short, the inspections observed during this study were different from the process described by Gilb and Graham, but they were typical, at least in this organization. Since the goal of our study was only to observe communication, these differences are not important. Rather, the organizational relationships and the types of communication in the meetings are the key aspects of this work.

The major dependent variable in this study is the length of the inspection meeting. In addition, dependent variables were defined to record the amount of time spent in each of five different types of discussion that took place during inspection meetings. The *defect* discussion time associated with an inspection meeting consists of time taken raising, recording, and discussing actual defects. *Global* discussion time includes discussion of issues applicable to other parts of the system as well as the code being inspected. Since this includes the raising and discussing of "global" defects, this category overlaps with the defect discussion category. *Unresolved* discussion time refers to discussion of issues that could not be resolved during the meeting. *Administrative* time includes time spent in administrative activities as well as the discussion of administrative or process issues. *Miscellaneous* discussion time includes miscellaneous discussions of a technical nature, including raising and discussing questions about the code being inspected that are not determined to be defects.

The initial breakdown of discussion categories was designed after a few preliminary observations, was modified slightly during the course of the study, and was collapsed (by combining some categories) to simplify the analysis. The breakdown presented above is the final breakdown used for analysis. An effort was made to make the categories "natural," especially to the project participants who were being interviewed. For example, the interviewees had no trouble understanding what was meant by the "time spent discussing defects" during the inspection meeting. Aside from the overlap between "defect" and "global" discussion time, the categories are mutually exclusive. This overlap is an artifact of the collapsing process. Originally, there were categories for discussion of "global defects" and discussion of "nonglobal defects," as well as "global" discussion time. None of the analysis performed is dependent on these categories being mutually exclusive.

One other dependent variable, preparation time, is the total amount of time inspectors reported spending in individual inspection of the material in preparation for the inspection meeting.

2.2 Organizational Structure

We have defined organizational structure as a network of organizational relationships among members of an organization. This network includes many different types of relationships. The independent variables in this study are measures of the different types of relationships that make up the organizational structure.

For example, one such relationship type consists of management, or reporting relationships. The independent variable *organizational distance* is based on management relationships. Each inspection is either organizationally "close" (all the participants reported to the same CSC manager) or organizationally "distant" (at least one participant from a different management area was present). Our understanding of the development project's management structure came initially from the *Flight Dynamics Division Phonelist*, which serves as an organization chart for FDD, and a less formal phonelist, which served the same purpose for the project personnel at CSC. This information was updated and validated through interviews.

Another independent variable, *physical distance*, reflects the number of physical boundaries between the inspection participants. In this study, physical distance takes on three values, corresponding to a set of inspection participants with offices on the same corridor, in the same building, or in separate buildings. FDD and CSC personnel are housed in different buildings, about two miles apart. As it turned out, in this study, physical distance was very highly correlated with organizational distance. Because of this, many of the findings in Section 4 hold for both organizational and physical distance.

Two measures of "familiarity" were used in this study, both based on pairs of inspection participants, and both ratio-valued. *Present familiarity* reflects the degree to which the participants in an inspection interact with each other on a regular basis, and thus presumably share common internal representations of the work being done. The value of this variable is the proportion of *pairs* of participants in the set of inspection participants who interact with each other on a regular basis. *Past familiarity* reflects the degree to which a set of inspection participants have worked together on past projects. This is assumed to contribute to a shared internal representation of the application domain in general, and a shared vocabulary. Past familiarity represents the percentage of pairs of participants who have worked together on past projects.

3 Research Methods

The methods employed in this study were both quantitative and qualitative. Data collection was largely qualitative, and data analysis employed methods of both types. A similar, but smaller, pilot study [19] was conducted in order to test this combination of methods and techniques. The design of the study described in this article is based on the design of the pilot study and the lessons learned. All of the methods and techniques used are discussed in the following sections.

3.1 Data Collection

3.1.1 Prior Ethnography

Before the data collection period, in November and early December of 1995, the first author conducted a *prior ethnography* by attending team meetings, conducting open-ended interviews with several developers and managers, observing several inspection meetings (without recording data), and being introduced to all project participants.

During this time, the development project was modeled using a formalism called *Actor-Dependency models* (AD models) [21], which also allowed us to automate some of the data analysis. AD models depict any system as a set of actors (human or automated) and the dependencies between them. Dependencies are of several types, including task and resource dependencies. For our purposes, the actors in the model were developers and other participants in the inspection process. The various steps in the inspection process were modelled as task dependencies, while the types of communication that took place were resource dependencies (based on the idea that information is a resource). AD models are implemented using Telos, an object-oriented knowledge base management language. This allowed attributes to be attached to actors and dependencies and values to be assigned to those attributes corresponding to the data being collected in the study. This greatly facilitated the management and analysis of data.

3.1.2 Documents

The official documents of an organization are valuable sources of information because they are relatively available, stable, rich, and nonreactive, at least in comparison to human data sources [14]. The AD model of the project environment relied heavily, at least initially, on process and organizational documents. The organization's inspection data collection documents were also employed. Further background information was provided by the online newsgroup used by the project personnel (one of the authors had direct access to this newsgroup). The drawback to relying on documents is that they cannot be assumed to be current, correct, and complete. For this reason, we used a variety of other information sources (in particular observations and interviews) to verify document-based information.

3.1.3 Observations

Much of the data for this study was collected during participant observation[1] of 23 inspection meetings between December, 1995, and April, 1996. Each inspection normally covered 2-3 C++ classes. The material inspected during the data collection period constituted over half of the system being developed. The time period chosen was based on the resources available to conduct the study and the schedule of the larger research program. All inspections which took place during this time period were observed. The idea behind observation as a research method is to capture first-hand behaviors and interactions that might not be noticed otherwise.[2] Information recorded during observations included administrative information, names and roles of each inspector, the amount (in LOC) of code inspected, and information about individual discussions. For each discussion that took place during the meeting, the observer recorded its length, its participants, the type of discussion (based on the previously mentioned categories), some notes indicating the topic and tone of the discussion, and any other relevant information. This data was used to calculate values for the discussion time variables described in Section 2.1 and shown in Tables 1 and 2. Extensive field notes were also written immediately after each meeting.

The accuracy of the information recorded relied on the observer's ability to determine when one discussion ended and another began, and to use the discussion categories consistently. Audiotaping the meetings was not feasible, so

1. "research that involves social interaction between the researcher and informants in the milieu of the latter, during which data are systematically and unobtrusively collected" [20].

2. Although the name is misleading, participant observation does not necessarily imply that the observer is engaged in the activity being observed (e.g., [1]) only that the observer is visibly present and is collecting data with the knowledge of those being observed. In this study, the observer did not play a role in the inspection process, but was visible to all participants.

in order to check the reliability of this information, a second observer was used in about 15 percent of the observations. The second observer, like the primary observer, was a computer science doctoral student with some work experience in the application domain. However, the second observer was unfamiliar with the development project being studied. The primary and second observers' records of the duration and participants in each discussion generally agreed. However, their coding of discussion types agreed only 62 percent of the time. In order to ensure that the coding was not completely subjective, an effort was made to resolve all discrepancies to the mutual satisfaction of the two observers, which we were able to do in all cases. The discrepancies were mainly due to the second observer's lack of familiarity with the project and with the primary observer's taxonomy of discussion types. The process of resolving these discrepancies resulted in one change to the taxonomy of discussion types, the addition of a miscellaneous category.

3.1.4 Interviews

The other important data source was a set of semi-structured[3] interviews [14] conducted with inspection participants. Each interview started with a specific set of questions, the answers to which were the objective of the interview. However, many of these questions were open-ended and were intended for (and successful in) soliciting other information not foreseen by the interviewer.

The interviews provided information about organizational relationships (particularly familiarity), data on inspection activities other than the meeting, and information on the code inspected (in particular, the interviewee's subjective assessment of its complexity). In some cases, the more straightforward questions were asked via email to reduce the amount of time the project personnel had to spend in interviews. Most interviews were audiotaped in their entirety. Extensive field notes were written immediately after each interview. The tapes were used during the writing of field notes, but they were not transcribed verbatim. For those interviews which were not audiotaped (because of reluctance on the part of the interviewee, or because of mechanical failure), notes were based on handwritten notes taken during the interview and on memory.

3.2 Data Analysis

Initial qualitative analysis of the data began about halfway through data collection. The first analysis was similar to the "constant comparison method" described by Glaser and Strauss [11] and the comparison method suggested by Eisenhardt in [6]. The method consisted of a case-by-case (meeting-by-meeting) comparison in order to reveal patterns among the characteristics of inspection meetings. The goal of this initial analysis was to suggest possible relationships between variables. These suggested relationships would then be further explored quantitatively where appropriate.

Some of the raw qualitative data was coded into quantitative variables, which fall into three categories. First are the dependent variables, which include the length of the inspection meeting and the amounts of time spent in different types of discussion, as described in Section 2.1. Secondly, there is a set of independent variables that represent organizational issues, as described in Section 2.2. Finally, there are two intervening variables, that must be taken into account so that the relationships between independent and dependent variables will not be masked by them. *Size* of the inspected material was provided on the inspection data collection forms, and was coded into a three-level ordinal variable for analysis purposes. *Complexity* of the inspected material was based on subjective interview data. Interviewees were asked how difficult the material was to inspect, and also what factors contributed to that difficulty. What emerged was not only an assessment of each set of inspection materials, but also a description of what contributes to complexity and difficulty, as perceived by inspectors. This is described in detail in [18]. Complexity was originally coded on a five-point ordinal scale, but was collapsed down to three levels in order to leave an adequate number of data points in each level.

The quantitative analysis used in this study was fairly simple and straightforward. We began by looking at descriptive statistics (mean, minimum, maximum, median, standard deviation) for each of the dependent variables and simple distributions of the independent variables. This helped to form an overall picture of the scope and shape of the data. Then we calculated Spearman correlation coefficients to determine which variables were statistically related (especially which organizational characteristics were related to measures of meeting length).

Qualitative data and findings were also used to help illuminate and explain the statistical findings. This was done in a more ad hoc way, by simply searching the field notes for anecdotes or quotes that shed some light on a particular finding. As well, after the initial quantitative results were generated, they were presented to several key developers on the project.[4] The developers' responses to and explanations for those results were recorded qualitatively and also helped illuminate the statistical results.

4 Results

Fig. 2 depicts the network of relationships among our independent, dependent, and other variables, according to the findings of this study. Each box represents a study variable or some other factor that became relevant during the course of the analysis. Each arrow represents some sort of relationship between two variables (e.g., a correlation) that was found in the data. Although causality is not supported by the statistical analysis performed, it is hypothesized in many cases (and obvious in others) in the directions implied by the arrows in Fig. 2. Some of the variables (those in dashed boxes) exhibited no relationships with any other factors. We discuss these variables and relationships in detail in the following sections.

3. A structured interview is one in which the questions are in the hands of the interviewer and the response rests with the interviewee. In an unstructured interview, the interviewee provides both the relevant questions and the answers [14].

4. This technique is called *member checking* [14].

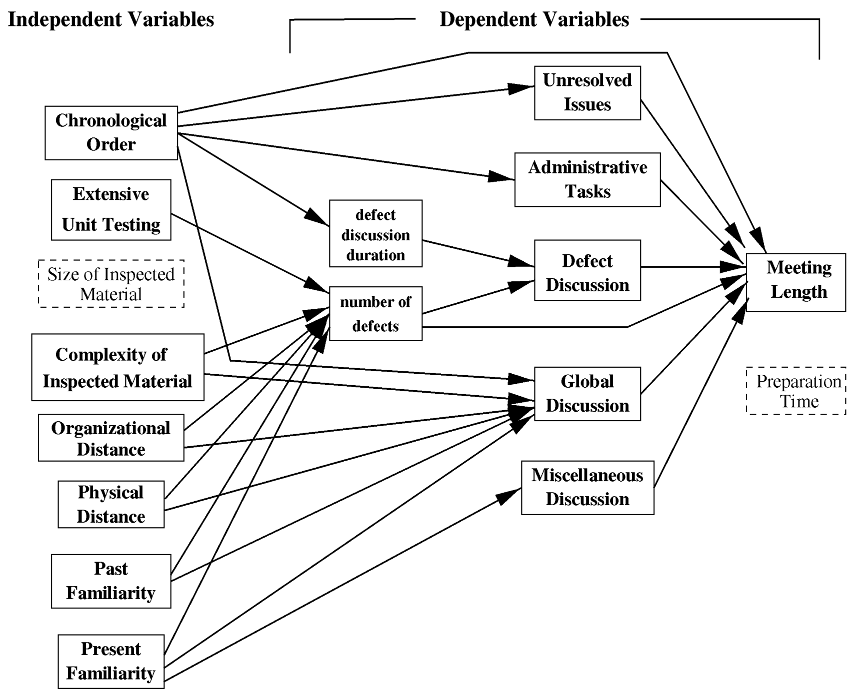


Fig. 2. The network of end.

In the discussion below, we first examine two subsets of meetings, the longest ones and the shortest ones. Focusing on these subsets accentuates the differences between long and short meetings, especially with respect to the types of discussions that comprise them, as depicted in the rightmost half of Fig. 2. Then we discuss the organizational and other factors that affect the amounts of effort spent in different types of discussion in all meetings. These relationships are depicted in the leftmost half of Fig. 2.

4.1 Components of Meeting Length

One useful way to view data is to look at its extremes. Tables 1 and 2 show the distributions of different discussion types in "long" and "short" meetings.[5] In both tables, the first row represents the six shortest meetings, all of which were 15 minutes or less in length. The second row represents the six longest meetings, all of which were at least 45 minutes in length. These categories represent natural breaks in the data. The figures in the first two rows of Table 1 show the average number of minutes spent in each type of discussion over the six meetings in each group. The third line shows the difference in the means for short and long meetings. In Table 2, the figures represent percentages of meeting time spent in discussions of each type, and the third line shows the differences in percentages. As explained earlier, the discussion categories are not mutually exclusive, but none of the analysis is affected by this.

Interpreting this data is not straightforward. From Table 1, we can see that the difference in the amount of time spent discussing defects accounts for nearly half of the difference between long and short meetings (23.1 out of 49.3 minutes). As shown in Table 2, longer meetings not only included a lot more defect discussion time, but a higher percentage of defect discussion time as well. Also, although time devoted to all discussion types is strongly correlated with meeting length, defect discussion time is the most strongly correlated. In other words, much of the extra time spent in a long meeting, as compared to a short meeting, is due to extra time spent discussing defects.[6]

Other discussion types also play a role in determining the length of an inspection meeting. Tables 1 and 2 show that the amount of time spent discussing unresolved and global issues increases for longer meetings, as does the percentage of meeting time devoted to such discussions.

From Table 1, we see that the time spent in administrative discussion (filling out forms, recording defects, etc.) was greater in longer meetings, but the difference is much smaller than with other discussion types. Table 2 shows that the percentage of time spent in administrative discussion is much larger in short meetings. This implies that the time spent in administrative tasks in an inspection meeting stays relatively constant and is nearly independent of the meeting length, at least as compared to other discussion types.

The last discussion category is miscellaneous discussion. As with other discussion types, longer meetings included more miscellaneous discussion time than shorter meetings. However, this type of discussion constituted a smaller percentage of longer meetings. Nevertheless, it accounts for a considerable amount of the difference in total meeting length between shorter and longer meetings.

It should be noted that the averages for the middle group of meetings (those between 15 and 45 minutes in length) show no anomalies or "spikes". That is, they lie between the averages shown in Tables 1 and 2 for long and short meetings. A similar analysis was also performed by dividing all the meetings into a long group and a short group,

5. The meetings ranged from 6 to 100 minutes in length; the median was 28 minutes and the mean was about 33 minutes. All of the data used in this study can be found in Appendix A.

6. Furthermore, more than four times as many defects were reported on average in long meetings than in short ones.

TABLE 1
AVERAGE AMOUNTS OF TIME SPENT IN DIFFERENT TYPES OF DISCUSSION AND TOTAL MEETING TIMES (MINUTES) FOR LONG AND SHORT MEETINGS

	Defect	Unresolved	Global	Admin	Misc	Total
Short meetings	3.7	0.7	1.0	3.7	4.2	12.7
Long meetings	26.8	6.3	9.0	8.7	13.8	62.0
Difference	23.1	5.6	8.0	5.0	9.6	49.3

TABLE 2
AVERAGE PERCENTAGES OF MEETING TIME IN DIFFERENT TYPES OF DISCUSSIONS FOR LONG AND SHORT MEETINGS

	Defect	Unresolved	Global	Admin	Misc
Short meetings	26	4	8	32	34
Long meetings	39	11	15	14	24
Difference	+13	+7	+7	–18	–10

based on the median meeting length. Similar results were obtained, but the differences between the two groups were sharpened by concentrating on the extremes, as described above.

This data paints an overall picture of the differing natures of short and long meetings. It appears that short meetings are dominated by administrative tasks and miscellaneous discussion. On the other hand, longer meetings are dominated by "weightier" types of discussion, particularly discussion of defects and global issues.

4.2 Effect of Organizational Factors

The previous section shows how long and short meetings differ with respect to the types of discussions that dominate them. But what is more interesting is how other factors, organizational and otherwise, affect the amount of time spent in different types of discussion.

To determine which organizational characteristics are relevant, we examined relationships between variables statistically using Spearman correlation coefficients. This test was chosen because it is nonparametric and does not require that the underlying distributions of the variables be normal (most of the study variables did not pass a test for normality). To examine the effect of the intervening variables, we calculated Spearman coefficients after partitioning the data by size and complexity. This was done to make sure that neither size nor complexity masked relationships between other variables. That is, we wanted to see whether or not certain relationships existed within subsets of the data defined by size or complexity.

The relationships found in the data are represented by the arrows in Fig. 2. It can also be seen from the figure that some of the study variables (in particular size and preparation time) did not demonstrate any relationships with any other variables. These are interesting results as well, and are elaborated below. First, we look at the number of defects reported in a meeting, and what organizational factors seem to affect it. The number of reported defects is interesting because it is strongly related to the amount of defect discussion time, which largely determines the length of an inspection meeting. Second, we look at the factors affecting global discussion time, which is another major determinant of meeting length. Finally, we present some observations relating to other discussion types.

It should be noted that defect data was not available for six of the inspection meetings observed. The findings related to the number of defects or the defect discussion duration are based on only 17 inspection meetings, instead of the 23 that comprise the whole dataset.

4.2.1 Reported Defects

The amount of time spent discussing defects during an inspection meeting is usefully broken down into two components, which the data shows to be affected by different variables. The first is the number of defects reported in the inspection meeting. "Reported defects," in this context, are defects raised during the meeting by one or more inspectors, discussed, deemed actual defects by consensus, and recorded on the inspection data collection forms. Thus, they do not include "false positives."

The second component is the "defect discussion duration," which is the average amount of time spent discussing each defect raised in a meeting. This component also exhibits considerable variation. However, surprisingly, it is unrelated to either size or complexity of the inspection material. In fact, it is not correlated, in general, with any of the study variables. Significant correlations were found only under certain conditions. For example, for material of medium complexity, the duration of *global* defect discussions decreased over time. That is, the later in the project that the inspection occurred, the less time was spent discussing each global defect (Spearman coefficient –0.81, $p < 0.05$). As discussed in the next section, this most likely has more to do with the global nature of those defects than any property of defects in general.

The number of reported defects, on the other hand, is related to many of the study variables. As might be expected, the defect discussion time is closely tied to the number of defects reported (Spearman coefficient 0.93, $p < 0.001$). It is also one of the few study variables (aside from the various discussion time variables) that is significantly correlated with meeting length (Spearman coefficient 0.72, $p < 0.005$), as can be seen from Fig. 2. Other variables have a more indirect effect, by influencing the amount of time spent in various types of discussion. This implies that the length of the meeting depends more on the number of defects reported than on any other factor. Other observations concerning reported defects are presented below. The observations (in this and the next section) are expressed as hypotheses, which could be investigated in future studies. Each proposed hypothesis is followed by the study findings that help support and interpret them.

Hypothesis: The more complex the material is, the fewer defects will be reported.

It is somewhat surprising that the number of defects reported in a meeting is statistically unrelated to the size of the material being reviewed. However, the other intervening variable in this study, complexity, did seem to have a moderate effect on the number of defects reported. Fewer defects were reported when complexity was high (Spearman coefficient –0.5, $p < 0.05$). It might be reasonable to assume that material of high complexity actually contained fewer defects (maybe because it was assigned to more skilled developers). However, another explanation is that inspectors had to spend more time understanding complex code, and thus did not have adequate time to search for defects.

Hypothesis: The more experienced or skilled the author is perceived to be, the less preparation time will be reported.

The result about complexity, discussed above, as well as other results reported later, raise an interesting issue concerning another of the dependent variables in this study, preparation time. The proposition that not enough time is given to inspecting highly complex material implies that the amount of preparation time expended is not determined by how much is needed to adequately inspect the material. The study data showed a wide variation in the amounts of preparation time reported (from 5 minutes to 26 person-hours), but preparation time exhibited no general statistical relationships with any other study variables. In particular, total preparation time was unrelated to either the size or complexity of the inspected material. However, dramatic differences were found when considering some specific characteristics of the author in each inspection. For example, preparation time decreased markedly when the author was one of the technical leads (although it should be noted that classes authored by the technical leads were significantly less complex than other classes). Also, preparation time was significantly higher for highly complex material when the author was having his or her code inspected for the first time. This strongly suggests that inspectors gauge the effort they are willing to expend for an inspection based on their beliefs about the experience or expertise of the author of the inspection material, not on the size or complexity of the code.

Hypothesis: The more familiar the inspection participants are with each other, the fewer defects will be reported.

When the material being inspected was of low complexity, fewer defects were reported when the inspection participants were very familiar with each other, based on either past or present working relationships (Spearman coefficients –0.95 and –1, respectively, $p < 0.1$). Also, no matter what the complexity, fewer defects were reported when the inspection participants were familiar based on past working relationships and the material inspected was small in size (Spearman coefficient –0.87, $p < 0.1$). So, for some types of material, closer past or present working relationships between the inspectors results in fewer defects reported.

This may indicate that developers are reluctant to report all the defects they find in material authored by close colleagues, or that they tend not to inspect such material as carefully as that authored by developers who are less familiar to them. The latter may be more likely, especially considering the stronger result for simple and small-sized inspection material. If inspectors are less likely to inspect carefully when the material is authored by someone familiar, then they may be even less likely to inspect carefully when, in addition, the material is small or relatively simple.

We also performed a side analysis to eliminate another possible explanation for this finding, that is that the familiarity measures also reflect the average experience of the inspection participants. That is, people who have been working (in the company or on the project) longer will be more familiar with more people. Thus, inspections involving more experienced people will also have higher values for the familiarity measures. This would imply that the fewer defects in the result cited above is actually an effect of experience, not familiarity. We ruled out this possibility by calculating two rough measures of average experience for each inspection (based on interview questions about length of participation in the company and on the project) and compared them to the familiarity measures, as well as to other study variables. The experience measures were the average number of months working for the company and on the particular project, respectively, over all the inspection participants. No significant correlations were found.

Another indication of familiarity is whether or not the author was in the "core group" which, for the purposes of this analysis, is defined as the eight developers who interact with other developers the most. This group consisted entirely of CSC developers, including the two CSC technical leads. All of the inspections included participants in the core group, but very few inspections involved exclusively core group members.

Inspections with a core-group author had less than half the number of reported defects than those with noncore group authors. When the author was one of the technical leads, the number of reported defects was less than a third than in other inspections. One developer explained this latter result by explaining that one of the technical leads is very experienced, and the other, although not very experienced, is "a whiz." The classes assigned to the technical leads also tended to be lower in complexity than other classes.

Hypothesis: When more unit testing is performed prior to the inspection, fewer defects are reported.

That extensive unit testing prior to the inspection reduces the number of defects reported in the meeting is intuitively logical. In two inspections, such extensive testing took place. In one of the inspections, one defect was reported, and two were reported in the other (much lower than the average of about 9). The low defect level cannot be explained by size or complexity. Because there were so few defects, there was very little defect discussion time, and the meetings themselves were correspondingly short. This result was especially satisfying to the developers to whom it was presented. Two developers, who both had leadership roles on the project, expressed the opinion that unit testing was a vital part of the development process, and this result was an indication that it was effective. However, since we do not know the actual defect densities of these classes, we might conclude that the inspectors may not have inspected as carefully because they knew that the classes had had extensive unit testing.

Hypothesis: The more distant the inspection participants are, either physically or in the reporting structure, the more defects will be reported.

Both organizational and physical distance played a role in one particular inspection with respect to the number of defects reported. This inspection meeting was an outlier, the longest in the data set, at 100 minutes. The author was an FDD developer, while all the inspectors were from CSC. This was an unusual situation in the data set. Consequently, the inspectors were not very familiar with the code before they had inspected it in preparation for the meeting. This meeting also had the highest number of defects reported in the data set, 42. This may have been partly a direct result of the high organizational and physical distance between the participants, particularly the author. Fourteen (compared to an average of 2) of the defects were global in nature, meaning that they were defects that had been raised in previous inspections. The author of the outlier inspection was physically and organizationally removed from the participants in those previous inspections. This may have contributed to a lack of communication about the global defects. This is consistent with remarks from developers, who described developers in other parts of the organization as “isolated.”

This outlier meeting suggests the following argument, which is consistent with the above findings on familiarity and distance. Different developers may be sensitive to different types of code errors, depending on their experience. Then the developers with whom an author consults during development (particularly those closer and more familiar) will help to eliminate certain types of errors from that author’s code. If those same developers are those who inspect that code, they may not find many errors because those they are most aware of have already been eliminated. But if a different set of developers inspects the class (who are more distant and unfamiliar), then they may bring different sensitivities to the inspection and thus find other errors (although they may take longer to do it). This may be what happened during the long outlier inspection. One developer addressed this very issue during an interview:

> She can imagine that if the inspectors are the same people who helped craft the code, then they’re not likely to find anything wrong with it. So this may be a reason to choose inspectors that are not that familiar with the code.

Unfortunately, these findings cannot be fully interpreted without knowing more about the error histories of the classes inspected. That is, we cannot know whether those classes that had fewer reported defects actually had fewer defects, or whether the organizational factors influenced the inspectors to find or report fewer defects than actually existed. It is important to look at this issue closely because the number of reported defects appears to have a very strong influence on meeting length. Thus it is important to know what factors affect the number of defects reported, besides the actual number of defects in the code.

For example, suppose we extrapolate the hypothesis about familiarity and the number of reported defects to imply that familiar inspection participants report a lower *percentage* of the defects that actually exist in the code. This is as reasonable a statement as any, as we have no reason to assume that the distribution of defects in classes inspected by a familiar group is any different from that of other classes. This would indicate that, while choosing a familiar set of inspection participants would seem to make the inspection meeting less time-consuming, it would seriously degrade its effectiveness.

In summary, a large part of the variation in meeting length is accounted for by the amount of time spent discussing defects, which in turn is largely dependent on the number of defects reported. This finding is somewhat comforting because the main purpose of an inspection meeting is, usually, to discuss defects. The number of defects is related to nearly all of the study variables, under different circumstances. However, the defect discussion duration also plays an important part in the amount of meeting time spent discussing defects. Unlike the number of reported defects, the defect discussion duration does not seem to be affected in general by any of the organizational variables, but under certain conditions it seemed to decrease over the course of the project.

4.2.2 Global Discussion Time

The time spent discussing global issues (including global defects) in an inspection meeting was strongly affected by a number of factors, as can be seen from the proliferation of arrows pointing to it in Fig. 2. It was, in fact, related to many of the same independent variables that were related to the number of reported defects. Therefore, some of the hypotheses presented in this section are similar to those presented in the previous section.

> **Hypothesis:** The more familiar the inspection participants are with each other, the less time they will spend discussing global issues.

First of all, global discussion time tended to be lower when the inspection participants were very familiar with each other, based on past working relationships. This correlation was not particularly strong in general (Spearman coefficient –0.38, $p < 0.1$), but was stronger for inspections of small amounts of material or material of low complexity. Also for material of low complexity, there was a strong tendency for global discussion time to be low when the inspection participants currently worked together a great deal (i.e. when present familiarity was high, Spearman coefficient –0.9, $p < 0.05$). In other words, people who interact on a regular basis spend less time discussing global issues only when the material being inspected is not very complex, but past working relationships have a more general effect. One developer addressed the latter result by observing that coding standards (which were the subject of many of the global discussions) are similar on all projects at CSC. So people who have worked together on past projects have most likely worked through some of these global issues together before, and thus it takes them less time to discuss them in the present. Also, it may be that developers are likely to discuss such issues outside the meeting with inspectors with whom they have worked before, thus reducing the need to discuss them during the meeting.

> **Hypothesis:** The more complex the material being inspected, the less time will be spent discussing global issues.

The above results are more interesting when we consider that complexity also had a direct relationship with global discussion time in the dataset as a whole. In general, the more complex the material, the less time was spent discussing global issues (Spearman coefficient –0.58, $p < 0.005$). This may indicate that, with highly complex material, the available time was spent discussing weightier issues than global defects, which are often "cosmetic." However, the results presented above on familiarity indicate that *more* time was spent in global discussion of simple material when the participants were familiar with each other. In other words, the general effect of complexity on global discussion time was outweighed by the effects of familiarity.

One implication is that the discussion of global issues is part of the learning process. It is during this type of discussion that general knowledge is shared. Participants who work together a lot, or who have worked together in the past, have many opportunities outside of the inspection meeting to engage in this type of learning, so less of it takes place in the meeting.

> **Hypothesis:** The closer the workspaces of the inspection participants, physically, the less time they will spend discussing global issues.

> **Hypothesis:** The distance between inspection participants in the reporting structure has an effect on the time they will spend discussing global issues, but the effect depends on the size and complexity of the material being inspected.

There were some very specialized relationships between global discussion time and organizational and physical distance in some parts of the data.

For material of low complexity, there was a strong tendency for more time to be spent discussing global issues when the inspection participants were organizationally distant (Spearman coefficient 0.87, $p < 0.1$). A similar result holds for physical distance. However, the effect of organizational distance on global discussion time is very different when we restrict the data to inspections of large amounts of material. For such inspections, there was *less* global discussion time when the participants were organizationally distant (Spearman coefficient –0.64, $p < 0.1$). That is, more organizationally distant inspection participants spent less time on global issues when inspecting large amounts of material. These results are contradictory, and they imply that any effect that organizational distance has on the amount of global discussion time is overshadowed by the size and complexity of the material to inspect.

The low complexity result makes more sense and is exemplified in the outlier meeting mentioned earlier (the longest meeting, at 100 minutes). The distance measures for this meeting were high, and it also included a large amount of global discussion. Global discussion constituted 18 minutes of the inspection meeting, which was much higher than the average of about 4 minutes. The complexity of the material was low, and it was small in size. The major factor seemed to be the organizational and physical distance of the author. Below is an excerpt from the field notes:

> One of the reasons this inspection was so long was that every "global" issue that had been hashed over in previous inspections was hashed out here as well, even a lot of things that had already been taken care of in [the code generator]. However, they all seemed to be a surprise to [the author], who hadn't gotten any of this presumably because he's at [FDD].

> **Hypothesis:** The later in the project the inspection occurs, the less time will be spent discussing global issues.

Global discussion time also decreased over time to some extent, especially for material that was large or highly complex (Spearman coefficient –0.53, $p < 0.001$). This was explained by one developer as largely due to the role of the code generator, which was being developed concurrently. Many of the defects that were raised repeatedly in different inspections (i.e., global defects) were eventually remedied by implementing the fixes into the code generator. So, early in the project, a lot of effort was made to specify these problems and solutions carefully for the developers of the code generator, so that they would be implemented correctly.

4.2.3 Other Discussion Types and Factors

It was mentioned earlier that the number of defects reported in an inspection meeting is one of the few study variables that is correlated directly with meeting length. The only other variable that is so correlated is the chronology of the meeting. That is, inspection meetings tended to decrease in length over the course of the project (Spearman coefficient –0.45, $p < 0.05$). The reasons for this are probably numerous and related to many of the results already presented.

Miscellaneous discussion time does not decrease significantly over time, nor is it significantly related to size or complexity. However, one component of miscellaneous discussion time (the amount of time spent asking and answering questions about the code being inspected) tends to be lower when the inspection participants are familiar, based on present working relationships (Spearman coefficient –0.65, $p < 0.1$). As explained by one developer, people who work together a lot are simply used to communicating, so can relay ideas very quickly. They also tend to discuss many issues outside the meeting, so less time is spent on them in the meeting.

As mentioned earlier, the time spent in administrative tasks in an inspection meeting is relatively constant, regardless of the meeting length. However, it did decrease over time (Spearman coefficient –0.52, $p < 0.05$). This is largely due to the fact that much of the administrative time in early inspection meetings was spent in asking and discussing questions about the inspection process itself. Inspections were just beginning on this project, the inspection process document had just been released, and inspections were being performed differently for this project in several ways. Inspection process questions consumed up to 5 minutes of each of the first 10 (out of 23) inspection meetings observed. After that, process questions did not arise, and the administrative procedures became a "habit," as one developer put it. Even with the extra time in the early inspections, however, differences in administrative time between inspection meetings do not account for very much of the variance in meeting length.

In general, more meeting time was spent on unresolved issues early in the project than later (Spearman coefficient –0.49, $p < 0.05$). This trend occurred because, as one developer explained, developers at first made an effort to resolve every issue during the meeting, even if they eventually found they couldn't. However, they later came to recognize more quickly which issues were best referred to someone else.

4.3 Limitations

There are a number of characteristics of this study that prevent it from more completely addressing the research questions. The first, and most significant, is its size and scope. The findings are limited to one software development project, in one application domain, during one period of time. Furthermore, only one part of the development process, code inspection, was studied. It's not clear how representative the studied project is, but the authors have attempted to include in this article enough details for the reader to determine how similar it is to their own context. No claims are made about the generalizability of the findings.

Another limitation of the study is that none of the variables were controlled. The inspection process was not manipulated in any way, so the results depended on the outcome of the development process as it unfolded. For this reason, no causal inferences can be drawn from the findings. Only correlations are reported.

A more fundamental limitation of the study is the scope of the questions it addresses. The issues are limited to three types of organizational relationships, and only one aspect of communication (effort). Not studied were issues related to the quality aspects of communication, or the relationship between communication effort and quality. Furthermore, only the amount of effort was studied, not the benefits or drawbacks of different levels of effort. Hopefully, these questions will be addressed in future studies. Until we gain more understanding of them, however, very little can be said about how software development practices could be improved to make communication more effective, or organizational structures more efficient. For this reason, no such general advice is given in this article.

The scope of this study also excludes an evaluation of the inspection process itself. Characterizing the efficiency and effectiveness of the inspection process studied was not a goal of this work. Consequently, the study has many limitations in its ability to make such an evaluation. In particular, neither testing nor operational defect data are available for the inspected code, which makes it impossible to determine how many of the defects in the code were found in inspection, relative to the total number of defects present. The defect data is not available because, after this study was concluded, but before the inspected code was tested, a major requirements change necessitated the rewriting of that code. Only after the rewrite was the code tested, so defect data would have no relation to the defects found during the original inspections. Data from unit testing (which occurred before inspection) would have been somewhat helpful, but was never recorded and so is also unavailable. Also unavailable is information about the relative seriousness of different defects. This information was not recorded as part of the study, nor by the project's regular data collection mechanisms.

5 Conclusions

This paper describes an empirical study of code inspection meetings in a NASA-sponsored software development project. The relevant variables in this study were process communication effort (in particular the effort expended in inspection meetings, in general and in discussions of different types) and characteristics of the organizational structure (reporting relationships, familiarity, physical proximity). We found that several organizational characteristics have an effect on the amount of time spent in different types of discussions during inspection meetings. Below, we present our findings in the form of testable hypotheses, which are among the main contributions of this work.

First, we presented results that showed that two of the major factors that make longer inspection meetings longer are the time spent discussing defects and the time spent discussing global issues. Furthermore, the time spent discussing defects is mostly determined by the number of defects reported during the meeting. The following hypotheses represent the study findings that relate to factors affecting the number of defects reported:

- The more familiar the inspection participants are with each other, the fewer defects will be reported.
- The more distant the inspection participants are, either physically or in the reporting structure, the more defects will be reported.
- The more complex the material is, the fewer defects will be reported.
- When more unit testing is performed prior to the inspection, fewer defects are reported.

The first two of the above hypotheses point to the conclusion that developers will report fewer defects in material authored or inspected by other developers with whom they are "close" (in terms of organizational distance, physical distance, or familiarity).

Another factor in the length of inspection meetings is the time spent discussing global issues, or those issues that arise repeatedly and are relevant to the system as a whole, not just the code being inspected. This study indicated that the time spent discussing such issues is strongly related to the organizational relationships between inspection participants, as detailed by these hypotheses:

- The more familiar the inspection participants are with each other, the less time they will spend discussing global issues.
- The distance between inspection participants in the reporting structure has an effect on the time they will spend discussing global issues, but the effect depends on the size and complexity of the material being inspected.
- The closer the workspaces of the inspection participants, physically, the less time they will spend discussing global issues.
- The later in the project the inspection occurs, the less time will be spent discussing global issues.
- The more complex the material being inspected, the less time will be spent discussing global issues.

In general, it can be hypothesized that inspection participants who are "close" spend less time discussing global issues. This is likely due to several factors, including the amount of discussion that goes on outside the inspection meeting, the shared vocabulary that arises from familiarity, and a shared understanding of the actual issues that come up repeatedly. Because less time is spent in global discussion, a close group of participants also results in a shorter meeting. This says nothing, however, about the effectiveness of such a meeting.

Finally, we have generated one other hypothesis related to another measure of communication effort:

- The more experienced or skilled the author is perceived to be, the less preparation time will be reported.

No other factors, such as the size or complexity of the code being inspected, exhibited significant relationships with preparation time. Assuming that other factors *should* play a role in an inspector's determination of preparation time, this implies that they need more guidance in this process.

These hypotheses could all be tested in carefully controlled experiments that are designed for that purpose. The study described here provides some evidence of their validity.

This study peels back just one layer of understanding about the role organizational structure plays in the efficiency of inspection meetings. The findings indicate that, in order to achieve shorter inspection meetings, inspection participants should be chosen so that they are familiar with each other, work in close proximity, and belong to the same organizational unit. However, there are also indications that such an arrangement results in a less thorough inspection of the material. As with most exploratory studies, this one raises as many questions as it answers. For example, What makes an inspection efficient? Is an efficient inspection meeting necessarily shorter? Does it necessarily report more defects? Does it have less discussion of some types and more of another? The answers to questions like these depend, at least in part, on the goals and objectives of inspection meetings, which vary from project to project. If they can be answered for a particular project, then studies like the one described here can provide guidance as to the organizational factors that can be manipulated, or planned for, to meet the project goals.

The work described in this paper helps to enable a whole area of research. Further work in the effects of organizational structure on the productivity of development processes has potential for profoundly influencing the success of software development projects. This study not only illustrates one effective way of conducting such investigations, but also provides some hypotheses with which to begin.

APPENDIX A—RAW DATA

TABLE 3
DEPENDENT VARIABLES: NUMBERS OF MINUTES SPENT IN DIFFERENT TYPES OF DISCUSSION AND TOTAL MEETING TIME IN MINUTES

Time Spent (minutes)					Total
Discussing Defects	Discussing Unresolved Issues	Discussing Global Issues	Administrative Tasks	Miscellaneous Discussion	Meeting Time (minutes)
5	0	0	3	7	15
16	3	8	4	8	30
20	0	3	7	11	36
18	6	0	2	14	40
0	0	0	5	5	10
1	7	0	1	8	17
1	0	1	3	2	6
6	7	2	1	11	25
12	0	1	7	2	21
4	0	2	5	10	19
28	5	8	4	5	41
2	3	2	2	10	17
13	7	7	4	18	45
6	0	1	3	6	15
7	0	3	2	4	15
25	4	13	8	9	54
16	4	9	5	3	30
66	3	18	14	17	100
3	4	1	6	1	15
4	10	3	6	5	28
9	5	8	10	12	46
27	8	3	5	16	66
21	11	5	11	11	61

TABLE 4
INDEPENDENT VARIABLES: PRESENT AND PAST FAMILIARITY, PHYSICAL AND ORGANIZATIONAL DISTANCE

Familiarity Based on		Physical Distance	Organizational Distance
Present Working Relationships	Past Working Relationships		
0.30	0.50	3	distant
0.33	0.00	2	close
0.30	0.10	3	distant
0.17	0.17	2	close
0.33	0.50	2	close
0.33	0.13	2	distant
1.00	0.33	2	close
0.33	0.33	1	close
0.33	0.67	1	close
0.33	0.50	2	close
0.67	0.17	2	close
0.17	0.50	1	close
0.40	0.30	2	close
0.60	0.20	2	close
0.20	0.20	2	close
0.40	0.07	2	close
0.10	0.10	3	distant
0.20	0.20	3	distant
0.50	0.17	2	close
0.27	0.13	2	close
0.33	0.20	2	close
0.10	0.30	2	close

TABLE 5
OTHER VARIABLES

Size of Inspected Code (LOC)	Complexity of Inspected Code	Total Preparation Time (person-minutes)	No. of Participants	No. of Reported Defects	Average Experience in Company (months)	Average Experience on Project (months)
10,000	very high	330	5	4	172	10
2,900	average	270	3	6	8	3
4,340	high	195	5	N/A	66	4
3,700	high	420	4	N/A	44	3
865	high	90	4	0	50	6
7,081	high	195	6	N/A	47	5
3,300	very low	90	3	2	33	9
5,251	very high	270	4	N/A	67	6
4,569	unknown	450	4	11	58	7
3,906	average	285	4	1	62	3
3,250	average	450	4	16	68	6
3,165	very high	690	4	1	67	7
4,924	low	200	5	6	84	10
4,100	average	135	5	5	32	10
5,454	average	510	5	12	51	6
4,392	average	5120	6	25	42	7
3,465	low	300	5	N/A	166	8
3,871	low	120	6	42	79	9
4,305	average	150	4	1	13	8
18,177	high	1,560	6	2	72	7
5,706	low	480	6	8	59	5
3,140	high	300	5	N/A	57	6
6,925	average	630	5	12	2	N/A

ACKNOWLEDGMENTS

The authors thank the members of the development team on the project used for this study for their patient cooperation. We would also like to thank the Experimental Software Engineering Group at the University of Maryland for their input at various stages of this work, including review of drafts of this paper. Carolyn Seaman's dissertation committee also provided valuable input to this work. Also invaluable were the comments from anonymous reviewers on previous versions of this paper. This work was supported, in part, by NASA grant 01-5-26393 and by IBM Canada Ltd.'s Centre for Advanced Studies.

REFERENCES

[1] S.R. Barley, "The Alignment of Technology and Structure through Roles and Networks," *Administative Science Quarterly*, vol. 35, pp. 61-103, 1990.

[2] K.V. Bourgeois, "Process Insights from a Large-Scale Software Inspections Data Analysis," *CrossTalk*, Oct. 1996.

[3] L. Briand, W. Melo, C. Seaman, and V. Basili, "Characterizing and Assessing a Large-Scale Software Maintenance Organization," *Proc. 17th Int'l Conf. Software Eng*., pp. 133-143, Seattle, Wash., Apr. 1995.

[4] B.G. Cain and J.O. Coplien, "A Role-Based Empirical Process Modeling Environment," *Proc. Second Int'l Conf. Software Process*, Berlin, Feb. 1993.

[5] B. Curtis, H. Krasner, and N. Iscoe, "A Field Study of the Software Design Process for Large Systems," *Comm. ACM*, vol. 31, pp. 1,268-1,287, Nov. 1988.

[6] K.M. Eisenhardt, "Building Theories from Case Study Research," *Academy of Management Rev*., vol. 14, pp. 532-550, 1989.

[7] M.E. Fagan, "Design and Code Inspections to Reduce Errors in Program Development," *IBM Systems J*., vol. 15, no. 3, pp. 219-248, 1976.

[8] J.R. Galbraith, *Organization Design*. Addison-Wesley, 1977.

[9] T. Gilb and D. Graham, *Software Inspection*. Addison-Wesley, 1993.

[10] J.F. Gilgun, "Definitions, Methodologies, and Methods in Qualitative Family Research," *Qualitative Methods in Family Research*. Sage, 1992.

[11] B.G. Glaser and A.L. Strauss, *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine Publishing, 1967.

[12] H. Krasner, B. Curtis, and N. Iscoe, "Communication Breakdowns and Boundary Spanning Activities on Large Programming Projects," G. Olsen, S. Sheppard and E. Soloway, eds., *Empirical Studies of Programmers*, second workshop, ch. 4, pp. 47-64, Ablex Publishing, 1987.

[13] R.E. Kraut and L.A. Streeter, "Coordination in Software Development," *Comm. ACM*, vol. 38, pp. 69-81, Mar. 1995.

[14] Y.S. Lincoln and E.G. Guba, *Naturalistic Inquiry*. Sage, 1985.

[15] J.G. March and H.A. Simon, *Organizations*. New York: John Wiley & Sons, 1958.

[16] E.E. Perry, N.A. Staudenmayer, and L.G. Votta, "People, Organizations, and Process Improvement," *IEEE Software*, pp. 36-45, July 1994.

[17] G.L. Rein, "Organization Design Viewed as a Group Process Using Coordination Technology," Technical Report CT-039-92, MCC, Feb. 1992.

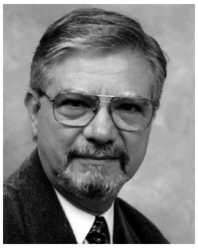
[18] C.B. Seaman, "Organizational Issues in Software Development: An Empirical Study of Communication," PhD thesis, Univ. of Maryland at College Park, 1996.

[19] C.B. Seaman, "Communication Costs in Code and Design Reviews: An Empirical Study." *Proc. CASCON'96*, CD-ROM version, Toronto, Canada, Nov. 1996. IBM Canada Ltd. Laboratory Centre for Advanced Studies and National Research Council of Canada.

[20] S.J. Taylor and R. Bogdan, *Introduction to Qualitative Research Methods*. New York: John Wiley & Sons, 1984.

[21] E. Yu and J. Mylopoulos, "Understanding "Why" in Software Process Modeling, Analysis, and Design," *Proc. 16th IEEE Int'l Conf. Software Eng.*, pp. 159-168, Sorrento, Italy, 1994.

Carolyn B. Seaman received her PhD degree from the University of Maryland in 1996; her MS from Georgia Insititute of Technology in 1987; and her BA from the College of Wooster, Ohio, in 1986. Dr. Seaman is a faculty researcher in the Experimental Software Engineering Group at the University of Maryland at College Park. She has conducted empirical research in various software engineering issues, such as the organizational structure of development groups, COTS use in development, technology transfer, and software maintenance. While working on her doctorate, she received a fellowship from IBM Canada's Centre for Advanced Studies. Previously, she worked as a software developer in a number of different environments.



Victor R. Basili is professor of computer science at the University of Maryland, College Park; the executive director of the Fraunhofer Center—Maryland; and one of the founders and principals in the Software Engineering Laboratory, a joint venture between NASA Goddard Space Flight Center, the University of Maryland, and Computer Sciences Corporation, established in 1976, recipient of the first Process Improvement Achievement Award by the IEEE and the SEI in 1994, and a NASA Group Achievement Award in 1996. He received the 1997 Award for Outstanding Achievement in Mathematics and Computer Science from the Washington Academy of Sciences. Dr. Basili has authored over 130 journal and refereed conference papers. He has served as editor-in-chief of the *IEEE Transactions on Software Engineering*, and as program chair and general chair of the Sixth and 15th International Conferences on Software Engineering, respectively. He is co-editor-in-chief of the *International Journal of Empirical Software Engineering* (Kluwer). Dr. Basili is a fellow of the IEEE and the ACM, and a golden core member of the IEEE.