Conceptual Simplicity Meets Organizational Complexity: Case Study of a Corporate Metrics Program

James D. Herbsleb and Rebecca E. Grinter
Bell Laboratories, Lucent Technologies
1000 East Warrenville Road
Naperville, IL 60566-7013
+1 630 713 1869
{herbsleb, beki}@research.bell-labs.com

ABSTRACT
A corporate-wide metrics program faces enormous and poorly understood challenges as its implementation spreads out from the centralized planning body across many organizational boundaries into the sites where the data collection actually occurs. This paper presents a case study of the implementation of one corporate-wide program, focusing particularly on the unexpected difficulties of collecting a small number of straightforward metrics. Several mechanisms causing these difficulties are identified, including attenuated communication across organizational boundaries, inertia created by existing data collection systems, and the perceptions, expectations, and fears about how the data will be used. We describe how these factors influence the interpretation of the definitions of the measurements and influence the degree of conformance that is actually achieved. We conclude with lessons learned about both content and mechanisms to help in navigating the tricky waters of organizational dynamics in implementing a company-wide program.

Keywords
Software Metrics, Technology Transfer, Organizational Dynamics

1 INTRODUCTION

Corporate-wide metrics programs provide unprecedented opportunities to gain insight into the software development process from hard data collected across divisions, products, and sites. Corporate-wide metrics programs also provide unprecedented opportunities to run aground on the complex realities of large corporations, as this centrally planned activity flows unevenly toward implementation in diverse organizations. The purpose of this paper is to provide a case study detailing many of these organizational dynamics, i.e., the paths and obstacles as they were revealed over a period of several years in one corporate-wide program.

In Section 1, we review some of the factors in an emerging consensus achieving success in software metrics programs. We also discuss the challenges of crossing organizational boundaries and the consequences of the fact that "software metrics" actually includes the quantitative component of several distinct organizational systems. In section 2, we describe the organization that was the subject of our study, and the methods we used to gather and analyze qualitative data. In section 3, we present our results, followed by a discussion and lessons learned in section 4.

1.1 Success in Corporate-wide Metrics Programs

There have been a number of published examples of successful corporate-wide metrics programs. One of the most widely known is Hewlett-Packard's (HP) [4, 6]. In HP's case, as is typical of such programs, the collection and use of software metrics data occurs at several levels in the organization, and serves several quite distinct purposes. At HP, for example, different kinds of data and analysis are used to manage projects, to evaluate products, and to improve the software development and maintenance processes [5].

As experience with these metrics programs accumulates, research has begun to form a consensus about some of the factors essential for success. An excellent summary of these success factors is embedded in Jefferey and Berry's measurement success factor framework [8]. These factors fall roughly into four categories:

- context, or overall environment of measurement effort,
- resources to support the measurement activities,
- process of the measurement activity, and
- products of the measurement activity.

One "context" factor, for example, is "There is senior management/sponsor commitment." An example of a "product" factor is "Constructive feedback on results is provided to those being measured." In a related paper, Offen and Jeffery [11] present their "model, measure, manage paradigm" which gives some guidance about putting these success factors into place, especially those calling for a close relationship between business needs and measurement.

Similar factors have been mentioned by other metrics researchers including a recent paper by Hall & Fenton [7] which identified a consensus around eleven requirements for successful programs. In an impressive series of publications spanning a number of years, Grady [4-6] has made many of

the same points. Pfleeger has also reported similar conclusions based on experience with a corporate-wide program at Contel [12].

Since there is a reasonable degree of experience-based consensus on what to do, one might reasonably wonder why there is also a consensus that establishing an effective metrics program is very difficult and prone to failure. Our experience with corporate-wide programs suggests that a good knowledge of *what* to do is important, but the complex dynamics of a large corporation pose formidable obstacles to figuring out *how* to make it happen. It is a bit like the famous advice for success in the stock market – "buy low and sell high." Very good advice, if you can figure out how to implement it.

Unlike the vacuous "buy low, sell high" advice, the consensus about success in metrics is meaningful and hard-won. However, as in the stock market case, knowing what to do does not solve the entire problem. For example, to pick just one element of the emerging consensus, there seems to be near-universal agreement that it is important that the developers understand the purpose and motivation of the metrics program. There may be any number of implementation problems with this suggestion. Any message about the program is interpreted by many different individuals based on such things as their history with metrics, their relation with management, their perceptions of the motives of those delivering the message, their inferences about how likely the program is simply to go away, their role in the organization, their personal goals, their project goals, business unit goals, and so on. Each of these can result in a "received" message that bears little or no resemblance to the "sent" message, and in fact the "received" message may vary tremendously from one part of the corporation to another. Similar issues surround implementation of the other metrics program requirements.

In order to make progress on the "how" question commensurate with the progress on the "what" question, we need to achieve a better understanding of the ways in which organizational dynamics impact corporate-wide metrics initiatives. This case study attempts a step in that direction.

In the next two sections, we examine two well-known sources of difficulty we expected to figure significantly in the organizational dynamics of our case study.

1.2 Perils of Crossing Organizational Boundaries

One of the most pervasive problems in large-scale software development is that projects span many disciplines and groups. Communication and coordination across team and especially organizational boundaries are very difficult and error-prone [3]. Members of different organizations and people occupying different roles tend to perceive such things as requirements in different ways, colored by their role, experience, and training. It would be surprising if the same were not true of software metrics initiatives.

Crossing boundaries creates measurement problems in many fields besides software engineering. A fascinating example is the seemingly simple process of recording causes of death in order to compile statistics for use in medical and public health research [2]. In practice, the picture is clouded by a number of factors, including different theoretical views about the role of various disorders in bringing about death, practical problems such as the availability of doctors to make the determination, local bureaucratic structures, laws and regulations concerning death certificates, and international politics. In one particularly interesting example, it is now thought by many public health researchers that the reported low rate of fatal heart attacks in Japan has little to do with disease processes. Rather, heart attack is a "low status" cause of death in Japan, connoting "a life of physical labor and physical breakdown" [2] (p. 75). Other, more "acceptable" causes of death, such as stroke, are often reported instead. If true, of course, this casts serious doubt on studies that have tried to identify nutritional or environmental factors in Japan that reduce the risk of heart disease.

Nearly everyone involved in software metrics will, I suspect, be able to think of analogous issues in software engineering. There are often local pressures, either real or perceived, to make some numbers appear low, and others appear high. Even in the absence of such pressures, interpretation of definitions often depends importantly on what the implementers infer to be the purpose of collecting a particular type of data. The potential for all kinds of misunderstandings is likely to increase substantially when data reporting crosses organizational boundaries. Among the reasons are

- Communication is attenuated across boundaries [3], and people draw inferences and generate hypotheses about the "real" purpose of measurement based on sparse information.
- Histories of past uses and abuses of data are likely to color interpretation of the present effort.
- Each organization has its own beliefs, values, and oral history that color the interpretation of all corporate initiatives.
- Appreciation of "practical realities," such as degree of disruption of existing systems and the "fit" of the new metrics with existing software processes, is attenuated across boundaries.

In the next section, we discuss several distinct, widely understood uses of metrics, and argue that reactions to the metrics program will be heavily influenced, often in unexpected ways, by perceptions of which system (or systems) will use the data.

1.3 Uses of Data in Several Organizational Systems

Organizations typically have many internal "systems" (see, e.g., [13]). "Systems," in this sense, include people, artifacts, and institutional arrangements that together carry out some essential function of the organization. These systems frequently need to gather information, communicate with other systems, plan and execute actions. Often, they

have a quantitative component so they can use corporate data to predict, understand, and influence the organization's behavior.

Distinctions among organizational systems are well recognized in the metrics literature, which often talks of metrics for several different purposes, most often including at least project management, software process improvement, and strategic management. There are, of course, other systems in each organization which are generally not considered part of the software metrics program, but that often collect similar data, such as payroll, effort reporting, defect tracking, and configuration management.

We talk here of "systems," rather than just "purposes," because it is important to realize that the characteristics of data which make it useful are determined by the characteristics of the entire system which will make use of the data. System characteristics such as the types of action taken by the system, the scope and time scale of action, and the degree to which it is reactive or proactive, all place requirements on the data it can use.

In addition to requirements on data that derive from the actual characteristics of organizational systems, the *perceived* relationship of the metrics effort to organizational systems will play a large role in determining how individuals react to it. As developers and metrics implementers reason about the corporate initiative, they are quite concerned with the motives and purpose of the program, as well as possible uses of the data and how the reported data will affect them. As we will see below, they tend to make assumptions about which organizational system will use the data, and these assumptions guide much of their thinking and behavior. Next we briefly review three of the major systems that are typical consumers of metrics data. (This description is highly simplified; fuller treatments can be found, e.g., in [4, 10]).

1.3.1 Project management system

The project management system is concerned with planning projects and tracking progress through delivery and maintenance. It can typically take actions such as reallocating project resources, renegotiating commitments, negotiating deliverables, or even canceling projects. It often uses quantitative analyses for such purposes as estimation and resource planning, monitoring product quality, monitoring resource expenditures, schedule adherence, and so on. In order to support these actions effectively, it must have data available very quickly and frequently during the life of the project. Consistency across projects is valuable, although not essential, because it allows accumulation of historical data to support estimation and establish expected ranges of variation. The data used are generally at the "subproject" level, such as effort expended to date, effort by phase, schedule variance, and so on. Obviously, a project management system cannot wait until data for the entire project are available.

1.3.2 Software process improvement system

The process improvement system attempts to change production processes to create better products more quickly and at lower cost. It can bring about incremental change, or it may redesign processes in a radical and discontinuous way. Quantitative analyses are used to identify process problems and bottlenecks, and to determine if changes have had the desired beneficial effects. This use of data generally assumes that process changes occur in the context of fairly stable common processes that are used across some set of projects, and that the data generated by those processes are comparable. Without these commonalities, relatively little can be learned about the characteristics of the process or the effects of changes. Subproject data may be useful, but interpretation is often ambiguous and reliance on such data can lead to serious suboptimization. Even if the requirements phase cycle time is successfully driven down, for example, unless the end-to-end cycle time is reduced, there is little or no benefit to the business. Depending on the nature of the improvement, data from multiple projects or an entire organization may be needed to avoid suboptimization.

1.3.3 Strategic management system

The strategic management system makes high-level long-term decisions to ensure the overall success of the business enterprise. It decides about such things as nurturing core competencies of the business, determining whether various functions should be performed in-house or outsourced, determining corporate policies and high-level resource allocation, as well as overall strategic direction about products and markets. Quantitative analyses are used for such purposes as identifying overall strengths, weaknesses, and competitive position. For these purposes, the system needs highly summarized data characterizing large parts of the organization, such as business units or the entire enterprise. It may also need industry data for purposes of benchmarking, goal setting, and making outsourcing decisions.

In some cases, the strategic management system establishes measures not so much to understand the characteristics of some process, but rather to influence it. When goals of, e.g., 10 times improvement in quality, or 30 percent reduction in cycle time, are set, this is done to pressure technical units to bring about a result, and even if not stated, it is assumed that consequences will attach to success or failure. Where this is the case, great care must be taken to insure the consistency and integrity of the data because of the enormous pressure to create the right appearance. One might look to accounting, with all its internal checks, auditing standards, and so on, for a model of how to actually generate data of sufficient quality for these purposes.

1.4 Goals for this study

Our overall goal in this case study is to try to shed some light on the organizational dynamics at work in one large corporate-wide metrics program. We focus particularly on the problems caused by crossing various organizational boundaries and the influence of perceptions about which organizational system will use the data. Specifically, we

try to answer the following questions:

1. As communications about the purpose of the metrics program cross organizational boundaries, to what extent and in what ways do these communications become attenuated?

2. In what ways does the limited upward flow of information about local environments impact the planning of the initiative?

3. In the face of limited information, how do metrics implementers and developers form their expectations, perceptions, and interpretations about the metrics program?

4. How do these expectations, perceptions, and interpretations influence the implementation of the metrics program?

5. What tentative lessons can we draw about the content and mechanics of a corporate-wide program?

2 EMPIRICAL METHODS, SITE OF THE STUDY

In this section we describe the organizational context in which the metrics initiative was unveiled, and the affect of the history of previous metrics programs on the present one. First we begin by describing the methods that we used to gather and analyze the data.

2.1 Methods Used in This Study

We chose a qualitative research method, since we were interested in learning about the motives, interpretations, actions, and context that influenced the way the metrics program was implemented. Data were collected using semi-structured interviewing techniques. Semi-structured interviews use a protocol — a series of questions designed to cover the range of topics — however, the questions are open-ended to encourage individuals to talk at length about their specific concerns. Using open-ended questioning techniques we were able to get information about the topics we were interested in and also learn about other related issues.

The protocol that we used for interviewing was designed on the basis of some initial conversations with key people involved in the metrics initiative. We also reviewed the extensive corporate metrics web site to learn about the definitions and tools used by the people collecting the data as well as the reported barriers and enablers of the metrics program. Furthermore, we attended meetings with the people responsible for implementing the metrics initiative. The combination of observation, interviewing, and artifact analysis helped to introduce us to the challenges of collecting and using metrics and design the protocol.

We conducted 15 interviews with people from 7 different projects. We interviewed 6 people involved in planning the corporate-wide program including the person responsible for starting the initiative. We also interviewed 11 people who collected metrics for specific products within the organizations, and had in that capacity been assigned the task of reporting the metrics for the corporate initiative. We chose our interviewees from diverse organizations, including at least one with a very sophisticated, long-standing metrics program, one that had not yet made much progress with metrics, and several at various stages in between. We also made sure that the projects sampled included a variety of products, and included a wide range of sizes, from less than a dozen people to several thousand.

All of the interviews except two were conducted with a single interviewee (the others had two interviewees). In each case there were two interviewers, one who took the role of ensuring that all the topics in the protocol were covered, and the other, freed from focusing on the protocol, was able to follow up on interesting but unanticipated remarks. To help ensure the reliability of the data gathered we also interviewed other individuals who had been involved with previous metrics initiatives in the same corporation. These people shared their own experiences and observations that substantiated, confirmed, and provided context for the patterns we found in the current initiative.

In addition to the notes taken during the interview, all interviews were tape recorded and transcribed. The notes and transcripts were initially examined for recurring themes. From this initial analysis we were able to devise categories of recurring themes. Using these categories we analyzed the interview materials, using the qualitative data to expand and revise our understandings of the systematic challenges in implementing corporate-wide metrics initiatives. To test our results we presented the findings to people involved in the metrics collection effort. They confirmed the accuracy with which we captured their interpretations.

2.2 The Metrics Initiative

2.2.1 The Ghost of Initiatives Past

This metrics initiative was not the first attempt to implement a corporate-wide program for software measurement. The corporation has had corporate-wide metrics programs at least two times prior to this. Due to lack of commitment, shifting corporate priorities, and changing markets, these older programs have been "lost" within the corporation.

For the people involved in gathering metrics for this newest initiative the older ones still play an important role. The older initiatives set a context for the implementation of *any* new corporate wide metrics program. Many of the people involved in the latest initiative remember the older ones from personal experiences of collecting data that often disappeared without a trace. We also heard descriptions of some uses of the data that the interviewees viewed as punitive. It was in this context, one that might be described as cynical or apathetic towards metrics, that the new initiative found itself being implemented.

2.2.2 The Current Initiative

The metrics initiative was one of a handful of initiatives adopted by a board consisting primarily of division-level managers whose organizations were heavily dependent on their software development and maintenance efforts. The initiative was given shape by a metrics requirements document prepared by two widely respected, senior members of the technical staff.

The initiative was relatively modest, straightforward, and simple in its design. Each organization was to report a small number (originally four, eventually expanded to a half dozen or so) of metrics. The metrics were defined corporation-wide, and were believed to be so basic that organizations would either already be collecting them, or would find this set a natural starting point.

The actual metrics to be collected were designed to be "end-to-end" metrics, focusing on the overall software process rather than more detailed metrics that focus, e.g., on phases of development. The initial metrics chosen were

- software size in function points,
- staff months of effort (which, with size, yields a productivity measure),
- interval, or cycle time from project go-ahead to market release, and
- high severity defects for the first 6 months after release.

To date, approximately 65 projects have reported at least one of these metrics. These projects are widely geographically dispersed, although they reside primarily in North America.

A metrics user group was formed as a vehicle for learning and sharing of knowledge. The user group had approximately 35 members, representing 15 different organizations. Meetings were conducted by conference call, generally about once a month, usually lasting about 2 hours. Meeting agendas typically covered current status of metrics collection as well as problems and issues with the collection and reporting. A web-based data repository was created, which contained all relevant documents, meeting minutes, and contact information, as well as the actual metrics data.

The primary purposes of the metrics, as reported in the requirements document, was for "use by business units to monitor the progress toward the goal of achieving Best-In-Class status for software development." The document suggests that improvement targets will eventually be set, and expresses the intent to make "software quality more visible." There is no hint that there is any intention to compare performance across business units, or even within a business unit (although we cannot rule out the possibility that comparisons were mentioned in early meetings about metrics).

3 EMPIRICAL STUDY RESULTS

We have organized our results into two major sections. First we look at the problems created by attenuated communication from the board to the implementers, about the purpose of the program and intended use of the data. As we will see, this attenuation did not produce a persistent information vacuum, however, as those involved with implementing the program used their prior experiences and their own understanding of metrics and of the organization to fill in the gaps. Both the attenuated communication and the interpretations it spawned had direct consequences on how the actual implementation was carried out.

In the second section, we examine the flow of information in the other direction, from local environments to the board. Details of these local environments, very difficult to anticipate from a central corporate perspective, led to unexpected difficulties in implementing an apparently simple set of metrics. There were also serious concerns in local organizations about supplying raw data, stripped of the context that would allow correct interpretation.

3.1 Which System Are We Dealing With?

Many of those responsible for implementing the metrics program expressed some confusion about its purpose. The initiative was shaped by a six-page requirements document (easily available on a well-publicized web page), but this document was not mentioned in any of the interviews. Not surprisingly, views about the actual purpose seemed to be formed primarily through informal channels and discussion. In this section, we examine the variety of conclusions reached by those responsible for implementing the metrics program and how these various conclusions influenced its implementation. While very few respondents expressed any certainty that they fully understood the intent of the program, they nevertheless actively hypothesized about its purpose (or lack of purpose) and made critical decisions about how to respond and how much effort to invest based on their conclusions.

3.1.1 Perception: Data will not be used by any system.

While several implementers of the metrics program said they really didn't know what the purpose was, others seemed confident that it actually had no real purpose, except perhaps that a metrics program looks good for public relations. One respondent remarked that "these initiatives often die because there *is* no purpose."

One common view was simply that the managers were trying to give a general boost to the idea of measurement, data collection, and the use of quantitative methods. By forcing everyone to get into the habit of collecting and reporting data, the effort would eventually spur thinking within each organization about how they might actually use data themselves. This "generalized priority boost" for quantitative methods in any or all of the organizational systems was mentioned favorably by those who held the view.

These views, not surprisingly, led to a strong desire simply to minimize the amount of effort spent supplying the corporate program with data. There was a general concern to be supportive of the effort, and to be as conscientious as they could under the circumstances, but compared to other "real" tasks that contributed to one's career and the success of the company, this effort was a low priority.

One consequence of the desire to minimize effort was that details of the definitions were generally not taken very seriously. Adhering strictly to the word of the definitions would have been very expensive in many cases, so data already available in other information systems, e.g., project management, defect tracking, and effort reporting systems, were typically entered (see section 3.2.1 below). It is

important to note that these variations were not motivated by laziness or the desire to subvert the measurement program. In the eyes of many metrics implementers, it was not clear that it was in their interest or the company's interest to invest the very significant amount of effort that strict adherence would require. After all, if the data were not going to be used, then their effort was better spent on other activities that would have tangible results. They were very aware of the opportunity cost of wasting their own time, a precious corporate resource.

Some implementers voiced a slight variation of this view, which was that the corporate metrics might benefit the company in some indeterminate way, but the cost would fall entirely on the business unit. In this "my business unit" versus the vague overall "corporation," the business unit was the clear winner. Again, there was a desire to support the corporate policy, but contributing to more immediate business success was a higher priority. Many voiced the opinion that if corporate wanted them to invest in corporate metrics, then corporate should provide the resources.

3.1.2 Perception: Data for Strategic Management System

Another common view was that the numbers would be used for strategic management purposes, specifically for such things as to see which projects were performing well, and to benchmark externally to see how the company stacked up against the industry in general.

Unfortunately, this view often took the form of a fear that the primary outcome would be to "beat up" projects or organizations that appeared ineffective because of e.g., high defect rates or low productivity. This is one sort of expectation typically generated by strategic management systems, in which goals are set and there is considerable fear about not "making the numbers."

For several of the interviewees, one important clue that caused them to believe the data would be used for comparative purposes was the very fact that uniform definitions were called for. Why else, the argument went, would the corporation try to establish common measures across many projects and organizations? Of course we all need a quality measure, but why do we all need to report exactly the same one unless there is an intention to compare?

One consequence of this view was a significant level of fear about how one project's numbers would look as compared to data from other projects. This impacted the actual implementation by influencing the way in which the inherent ambiguity in the measurement definitions was resolved.

There was considerable, and nearly unavoidable, ambiguity in the definitions. No matter how precisely definitions are worded, when they are applied in many different organizations with different processes, different products, release cycles, and so on, considerable interpretation is required. These issues were often raised and documented in the periodic teleconference meetings, but they were too complex to be resolved uniformly in this manner.

These ambiguities allowed individual organizations to create their own interpretations. Not surprisingly, they tended to choose interpretations that would produce numbers that "looked good." For example, in the definition of "customer found defects," it was not clear whether this included only defects actually reported by customers, or whether it included all defects found by anyone after release. Obviously, the narrower "actually found by customer" would produce a considerably smaller number and the implementers were quite aware of this. In each case where this issue was mentioned to us, the smaller number was chosen.

Similarly, there was some ambiguity in which defects were to be included in various counts, both in terms of how they were discovered (code inspections: yes; design reviews: unclear) and what precisely constituted severity 1 and 2, the only levels that were to be reported. Again, if the desire is to look good in terms of, say, defect removal rates, the definitions can be, and to some extent were, manipulated to achieve this. There was no sense that this was "cheating" or corrupting the system in any way, since the choices were viewed as essentially arbitrary, and it was expected that others would be similarly motivated.

3.1.3 Perception: Data for process improvement system.

A number of interviewees concluded that the metrics were likely to be used to support process improvement efforts. Although this potential use aroused less fear than strategic management uses, it was not generally seen as a legitimate use of these particular data. A frequently mentioned problem was, in the words of one interviewee, that "these metrics are not adequate to change anything."

Several concerns were raised about this use. First, the discretion each organization had in actually applying the definitions meant that interpreting differences between organizations was impossible. Just because one organization produced more "function points" per "man-month," or had fewer "defects" per "function point," most were unconvinced that this was at all informative about actual efficiency or quality. The numbers could be too easily manipulated, and too easily influenced by arbitrary decisions about how to interpret definitions. The second concern was that so many things were changing at once, including gaining experience, numerous quality initiatives, new tools, differences among features, differences among people, and so on, that it would be impossible to determine what *caused* differences in performance, even if they could be measured.

Another issue raised by those who anticipated a process improvement use of the data was that the measures selected were not sufficiently detailed for this purpose. Suppose, for example, that one project had particularly good quality numbers. Other projects could not determine, for example, if this was achieved by a low injection rate, good design reviews, especially effective code reviews, and so on. It was

not clear how overall, summary numbers would help anyone improve.

Finally, several interviewees mentioned that the data did not have the precision necessary to be used for process improvement. Effort data, for example, were often not recorded with great precision. If, for example, work on one feature of a product was going well and another was not, more effort might be devoted to the troublesome feature without this being reflected in how the time was reported. In some parts of the company, this is a very common practice, and was seen as maintaining a necessary degree of flexibility in shifting resources to where they were needed. To use such data to determine, e.g., that one process was more efficient than another would clearly be inappropriate.

One effect of these concerns was, once again, to minimize the effort expended on data collection and reporting. If the data cannot really be used for this perceived purpose, it makes no sense to invest heavily in data collection. This feeling was tempered, as in the strategic management case, with the desire not to "look bad." The fear here was that if their processes looked bad, they might be asked to adopt other processes that "looked good" according to the metrics, even though these appearances of "good" and "bad" were brought about, in their view, essentially by chance. This could force them to waste enormous effort.

3.1.4 Perception: Data for project management.
The perception that the corporate metrics would be used for project management was not widely held. This model still influenced perception of the metrics program, however. Individuals whose primary experience and interest was in such things as estimation and project tracking interpreted the corporate program in light of expectations generated by their experience.

Those viewing the corporate initiative through this lens complained that the corporate data would not be as useful as their own data because they had refined their estimation processes over a substantial period of time. They knew what data allowed them to make good estimates, and how to track projects against these estimates. Without this kind of iterative refinement and accumulation of historical data, the corporate program was not viewed as potentially helpful. In addition, the corporate data were not viewed as timely, since by the time they appeared on the corporate web site it would be too late to take any corrective action.

3.2 Through a Glass, Darkly

When corporate wide initiatives are implemented they must successfully negotiate all the challenges of communicating and coordinating across organizational boundaries. In this section we describe some of the problems that the metrics initiative faced as it crossed those boundaries. In particular, we examine first how the details of local environments, details that are very difficult to perceive from a corporate perspective, created unanticipated difficulties. Second, we examine the effects of the anticipated loss of contextual information, as numbers are reported without opportunity for explanation.

3.2.1 Interaction with Existing Tools and Processes
For many years, corporations have experimented with software metrics programs, and have in addition adopted different technologies to help them track defects, manage projects, and keep track of expenses in the development process. This corporation was no exception. A mixture of in-house and commercial systems for recording quantitative data was already in place. Despite the prevalence of existing tools, little was said about how to fit the new initiative to the existing technology. We found that the initiative was compromised by existing systems in a number of significant ways.

One of the corporate metrics reported was the effort that individuals spent building the software. Rather than implement a new way of collecting effort data, the implementers relied on the time keeping system. This system is uniform across the organization, and everyone knows how to record effort in terms of this system. These features make it attractive for metrics collection because it is widely adopted and well understood.

However, the time keeping system was originally intended for accounting purposes, specifically payroll. As a result the data reported were problematic for a number of reasons. First, the system did not handle any unpaid overtime. Unpaid overtime — especially during times when deadlines are approaching — can form a substantial amount of the actual effort of building software. Yet because the system was designed for payroll, it did not maintain information about unpaid time. Consequently, the effort went unaccounted for, even though all effort, including unpaid overtime, was to be included by the corporate definition in order to have data more suitable for planning purposes and for meaningful measures of productivity.

Another problem with the data that were reported by the system arose from the way data were typically entered. Depending on the organization, employees are required to fill out weekly or bi-weekly reports about where they spent their time. For a variety of reasons, there was great skepticism about the precision of these reports. In the first place, no one claimed to be able to recall where time was spent over an entire week or two weeks. In some cases, developers can charge their time to one development effort exclusively; however, in many cases the developers would be working on two or more different development efforts — often multiple releases of the same product. Since time reporting was often viewed as relatively unimportant, there was considerable skepticism about the accuracy with which people were dividing up their time.

Another type of widely used system was a mechanism for recording defects. Defect tracking systems can be part of configuration management systems, or exist on their own, and within the corporation there was a wide proliferation of both kinds. The different development groups had adopted systems that met their specific needs, and so with the advent of the metrics initiative they began to use the defect data in their systems to meet the demands of the new

program.

One clash that emerged was in the meaning of what defects were reported. Some systems used a scheme where they recorded the severity of the defect. The more damage the defect did on execution, the greater the severity. Severity was incorporated into the official corporate definition. However, other development groups were tracking defects based on priorities. Each defect was given a priority, the greater the priority the sooner the defect would get fixed. Often severe defects have a high priority. Sometimes though, high priority defects are not severe, instead they carry a high priority because of a customer demand to fix them, or some other business goal.

For the individuals collecting priority metrics, the new initiative presented them with an accuracy-versus-cost tradeoff that was not always resolved in the same way. There was significant extra work involved translating their defect priorities into severities so that they could report their data correctly. This was a manual process that required individuals' judgment on whether this high priority defect was also a high severity defect.

As was the case for tools and for locally adopted definitions, we also found that organization-specific software development processes constrained and shaped the implementation of the metrics initiative. Some of these processes are formally initiated and implemented. Others come from years of collected and shared experiences. At the corporation most of the processes have now achieved a formal status with the advent of ISO 9000 certification that requires the development processes to be documented. This certification creates inertia for changing processes, because although it can be done, it requires extra effort.

Changing individuals' or teams' processes also created problems for the corporate metrics initiative. The corporation — like many others — encourages individuals and teams to "own" processes for development. Over time this has led to a sense of responsibility for maintaining and changing the process in line with the demands of the project. A number of the people responsible for maintaining the metrics processes for their projects spoke to us about how they'd made refinements to make those processes more relevant to their projects. They had come to truly own their processes, and took pride in them.

The metrics initiative asked individuals, in effect, to change their processes, often removing those refinements for the sake of creating a common and uniform corporate definition. Had the individuals removed their changes and refinements from the process completely then the process would not have belonged to them anymore. Instead most of the people we spoke with generally chose to increase their workload by retaining their processes and doing the extra work required to distill the corporate data out. This mode of working required more time and effort, and also led to a sense that the corporate wide effort was not as sophisticated as their own data collection, and therefore somewhat pointless.

The cycle-time metric developed as part of the corporate initiative provides our final example of how the initiative was influenced by existing processes and tools. The cycle time metric that the board adopted was designed to fit into an overall end-to-end measure from the time when a customer suggested the change, or the development project was conceived, until it entered the marketplace. They approached this by adopting a stage-gate model which was used to manage and track the entire process, and requiring the software implementers to report the interval between the two stage gates that marked the beginning and end of the actual software development.

One major difficulty was that there were several different sets of "stage gates" already in use within the corporation. A number of development groups attempted to transition from their ways of measuring cycle time to the one adopted by the board, but could not adapt. These development groups found the approved measure difficult to fit to their projects, and the definitions that marked key points in the development interval difficult to apply to their circumstances. The were extremely skeptical that the essentially arbitrary mappings they were forced to construct would produce intervals that could be meaningfully compared.

Even groups already using the recommended stage gates had very different interpretations of them, and often took different stands on the ways that they completed the milestones. We found that some projects rushed to complete certain stage gates, while others waited much longer before passing the same milestone. One group viewed these gates as so meaningless, they reported a different interval instead, on the belief that it was a better reflection of their actual cycle time. Their integrity was admirable, since the substituted measure tended to produce longer cycle time measurements.

In this section, we looked at several ways in which the details of the local environment, difficult to perceive from a corporate perspective, generated unexpected difficulties with collection of the corporate metrics. In the next section, we examine how this inability to perceive the local environment created the expectation that data would be interpreted in simplistic and misleading ways.

3.2.2 Interpreting Data in Context

The interpretation of data also reveals the problems in communicating across boundaries. The data are a communication to others outside the local organization about the state of the project. What the data say about the project — and what they do not explain — are critical. In this section we review some of the problems that the interpretation of data presented to this initiative.

The individuals collecting the metrics were concerned about how the results were being interpreted in the absence of knowing the operating context of the project. In the local development projects individuals responsible for gathering and analyzing metrics bring their knowledge of the environment to bear on the results. For example, when

intervals reported are longer than desired, the individuals can generally point to events that happened to stretch the times. In one case the cycle-time for a project was exaggerated when an external equipment manufacturer delayed their shipment of a component. Since this happened when the project was between the stage gates that define the software interval, the developers feared that blame would be laid at their door.

When the metrics were reported, they were reported only as numbers entered in a form. There is no opportunity to provide the context that would make the numbers more meaningful, or to look for causes behind things like long intervals and schedule slips. Neither is there much opportunity to bring information about the local operating environment to bear on the interpretation of the metrics.

A major consequence is that those responsible for data collection generally believed that their data were likely to receive a simplistic interpretation. As one individual involved in data collection said, "there is not much subtlety to the communication."

As we conducted our interviews, we were surprised at the number of interviewees that spontaneously and quite favorably mentioned another, non-metrics initiative sponsored by the same board. In the next section, we try to identify the underlying mechanisms responsible for this favorable reception.

3.3 A Successful Non-metrics Corporate-Wide Initiative

In contrast with the metrics effort, the software process assessment (SPA) initiative was considered a success by all who spoke of it. During the interviews most individuals talked spontaneously about SPA and how useful it had been to them. In this section we outline some of the features of the SPA program responsible for its perceived success. We believe that the metrics program has much to learn from the assessments initiative, because the SPA provides a mechanism that makes it happen.

The SPAs were facilitated by a trained lead assessor from a corporate department that specializes in technology transfer. He trained the assessment team, which consisted primarily of local personnel. After the assessment, the team shared the results with the development group and their managers, and helped them understand the findings. Like a consultancy, the assessment team charged each development project for their services, so projects could have seen it as a cost, with little benefit, but they did not. The SPA was liked and used by different projects for a number of reasons. Individuals told us about several kinds of benefits.

The assessment includes members of the project that span all the functional areas involved in the development effort. This ensures that boundaries within the project are spanned, leading to a broader perspective on what is happening. The team also shares their results with the developers and managers in the assessed organization, people who can bring their knowledge of the local operating context to bear on the results.

Another benefit of this approach was the ability of the assessment team to share information among the different development groups, across projects and business units. The assessment team acts as a central point for information about conducting SPAs but also, importantly, about resolving problems raised from the results.

Finally, the assessment team provided results that were actionable, and were designed to meet the needs of the local organization. In some cases, these findings were seen as an enormous help in setting priorities and figuring out what needed to be done. In other organizations, with established quality teams, the results served primarily to tune or confirm the ongoing efforts. But the reception was uniformly much more favorable that the response to the metrics initiative.

In acting as a central resource the assessment team has accrued a significant amount of knowledge about the positive and negative parts of each development groups' development processes. The assessment team has also acquired the reputation of being able to share this information with other development groups in ways that leave all the parties involved feeling comfortable. The lead assessors are diplomats who connect groups: one that needs a solution from another; and to compare results with context, but without revealing where those results came from.

The assessment team spans three kinds of boundaries to make the SPAs work:

- boundaries among functional areas inside the project,
- boundaries among the projects sharing information about the kinds of improvements other projects are making — a kind of technology transfer, and
- boundaries between the projects and corporate management, to create an accurate picture about the state of process improvement, accounting for local context, and without violating confidences or creating a threatening atmosphere.

This comparison is strengthened by some of the comments made to us about frustrations with the metrics program. One person said, for example, "We'd really like pointers to help" and was frustrated that the corporate metrics effort didn't supply this. In a similar vein, another said, "Identifying organizations that do things well would be useful." The SPA initiative provides a model for accomplishing this without the threat of publicly posted lists of villains and heroes.

4 DISCUSSION – LESSONS LEARNED

There are several lessons that emerged from this study:

Meaningful organization-wide metrics may be prohibitively expensive, even when they appear simple, and these costs are likely to be grossly underestimated.

The difficulties in communication, the inertia from existing systems and processes, the inevitable ambiguities when definitions are applied in diverse contexts, and the difficulty

of establishing a perceived benefit all conspire against corporate-wide programs. The difficulties may not be at all obvious, however, since they are hard to see from a corporate perspective that necessarily glosses over the operational details of working environments. The difficulties may never be discovered, because numbers of some sort will probably be reported.

To collect high quality data would be, we believe, extremely expensive. The field of accounting, with its highly elaborated standards and practices, specialized training, internal checks, and regular audits, gives a model of what is required.

Bundling together the quantitative elements of several organizational systems as "a metrics program" is a flawed tactic.

Confusion about how data will be used is a pervasive concern in software metrics. Concern about the "human element" has been raised many times before, of course (see, e.g., [6] esp. ch. 7), generally in the context of trying to reassure people the data won't be used to hurt them. There have also been many cautions over the years to be sure that there is a match between data to be collected and questions to be answered [1]. While we agree in general with the advice one finds in the literature, we think it may be time to consider a slightly more radical approach. It might be wiser not to have a "metrics" program at all.

If the need is for better software process improvement, then have a software process improvement program that, of course, will have a quantitative component. If the need is for better project management, then, have a project management initiative, again, with an appropriate quantitative component. If there is a need for a better management information system, create one. We are beginning to wonder what advantage there would ever be to bundling these together as a "metrics program." The kinds of data needed differ considerably, as do the timeliness and precision requirements. As we saw, confusion over how the data will be used is an enormous problem.

The only advantage we see to the bundling together of the quantitative components is that it may make it easier to create a common infrastructure and toolset for data collection. This is an important consideration, but it is not clear that this advantage outweighs the confusion such programs seem to generate.

The importance of a boundary-spanning role should not be underestimated.

As in software development [3], the role of boundary-spanner seems also to be critical in other related activities. The spontaneous remarks of our interviewees indicated that the lead assessors in the SPA initiative played such a role and were highly valued because of it. It is worth considering ways of creating such a role in a metrics program.

One can imagine many ways to accomplish this. For example, Jefferey and Berry's measurement success factor framework [8], or a process for creating a "scorecard" [9] tailored for software, could serve as the "assessment analog." An internal consultant with broad experience and the ability to perform discrete boundary-spanning could provide the needed mechanism. Sponsoring such an activity might be more meaningful than attempting to define standard corporate-wide metrics.

ACKNOWLEDGMENTS

We would like to thank all of those who participated in the study. We would also like to thank Hal Lowe for his many insightful comments.

REFERENCES

[1] V. R. Basili & D. M. Weiss, "A methodology for collecting valid software engineering data," *IEEE Transactions on Software Engineering*, vol. 10, pp. 728-738, 1984.

[2] G. Bowker and S. L. Star, "Situations vs. standards in long-term, wide-scale decision-making" in Twenty-fourth Annual Hawaii International Conference on System Sciences, Hawaii, 1991.

[3] B. Curtis, H. Krasner, & N. Iscoe, "A field study of the software design process for large systems.," *Communications of the ACM.*, vol. 31, pp. 1268-1287, 1988.

[4] R. B. Grady, *Practical software metrics for project management and process improvement.* Englewood Cliffs, NJ: Prentice Hall, 1992.

[5] R. B. Grady, "Successfully applying software metrics," *IEEE Computer*, pp. 18-25, 1994.

[6] R. B. Grady and D. L. Caswell, *Software Metrics: Establishing a Company-Wide Program.* Englewood Cliffs, NJ: Prentice Hall, 1987.

[7] T. Hall & N. Fenton, "Implementing effective software metrics programs," *IEEE Software*, pp. 55-65, 1997.

[8] D. R. Jeffery and M. Berry, "A framework for evaluation and prediction of metrics program success," presented at First International Software Metrics Symposium, Los Alamitos, CA, 1993.

[9] R. S. Kaplan and D. P. Norton, "The balanced scorecard -- measures that drive performance," *Harvard Business Review*, Jan.-Feb., pp. 71-79, 1992.

[10] Software Engineering Laboratory, "Software Measurement Guidebook: Revision 1," Goddard Space Flight Center, Greenbelt, MD SEL-94-102, June 1995.

[11] R. J. Offen and R. Jeffery, "Establishing software metrics programs," *IEEE Software*, pp. 45-53, 1997.

[12] S. L. Pfleeger, "Lessons learned in building a corporate metrics program," *IEEE Software*, pp. 67-74, 1993.

[13] P. M. Senge, *The Fifth Discipline*: Doubleday, 1994.