Measurement: The key to application development quality

by C. Walrad
E. Moss

Application development quality and productivity have been identified as being among the top ten concerns of information systems (I/S) executives in both 1991 and 1992. This paper discusses the role of measurement in pursuit of I/S application development quality and productivity. The relationships between productivity, quality, and measurement are described, classes of measures are identified, and "dominant measures" are grouped according to the maturity levels defined by the Software Engineering Institute's Capability Maturity Model for Software. Also discussed are the organizational and cultural issues associated with instituting a measurement process.

The phrases "market-driven quality," "total customer satisfaction," and "total quality management (TQM)" often come up in executive conversations today because of the growing awareness that the ability of a business to compete effectively depends on increasing quality and productivity.[1] Many companies are actively re-engineering their business processes to achieve quality and productivity improvements. Information systems (I/S) technology often plays an important role in business process re-engineering initiatives. As a result, I/S customers throughout the enterprise are demanding higher-quality software applications that do more and are delivered faster. Software managers, like their line-of-business customers, are struggling to improve quality and productivity. Unfortunately, higher quality and greater productivity are often viewed as competing, rather than complementary, goals. Consider, for example, organizations that seek to improve application development quality through increased testing. These organizations usually find that quality improves, but overall development productivity suffers. The additional testing requires additional time and expense, both for the testing effort and for correcting the defects that testing uncovers.

The conclusion that quality improvements come at the expense of productivity is caused by not viewing the relationship between quality and productivity as part of a total system. By analyzing the software development life cycle as a whole, I/S organizations can look at ways to build quality into the whole system, from initial conceptualization of a software application, all the way through maintenance and obsolescence.

This paper is directed at I/S organizations that are considering the implementation of a software measurement process as a means to quantify and improve their value to the enterprises they serve. We hope that these organizations will benefit from an overview of measurement principles, a translation of these principles into I/S terms, an appreciation for the hierarchical nature of meaningful measures, and an awareness of the cultural and organizational aspects of deploying a measurement process.

Table 1 Quality dimensions and typical questions

Quality Dimension	Typical Question
Right Thing	• Are we sure I/S resources are deployed as effectively as possible? Why can't more effort be directed at delivering new applications? • How can you justify this budget request? It is over 20 percent higher than last year! • What are we getting for our computing dollar? • I have heard complaints from field sales about the order processing system. How satisfied are your I/S users overall? • What is I/S doing to support the business?
Right Way	• How flexible are we? What does it cost to modify applications when business processes change? • Do we have a well-defined software development process? How well are we transferring successful methods and technology across projects? • How well do we understand and capture requirements? • Did corporate priorities change or did emergency "rush" jobs disrupt previously scheduled activities? Did we force schedule adherence without meeting quality goals? • How can we work with the user community to improve our effectiveness? • What is the cost of our I/S products?

Quality drives productivity. Many experts in quality, including W. Edwards Deming, Joseph Juran, and Philip Crosby, have shown that the key to improving product quality lies in improving the quality of the process by which the product is made. Reduced rework and improved productivity are direct results of improving the quality of the production process. This conclusion is supported by a Federal Quality Institute finding in the mid-1980s that "productivity is a by-product, a result of quality improvement."[2] Software organizations are more frequently reporting that increased quality results in increased development productivity.[3]

If the concept that quality drives productivity seems obvious, why is it that many I/S TQM efforts are not underway? One reason may be that I/S has been traditionally viewed by an enterprise as an expense center. Because expense centers do not interface directly with external customers and do not usually produce products used by these customers, they are typically the last areas targeted by TQM initiatives.

A more fundamental reason, in our opinion, is that effective quality initiatives require clear definitions and explicit measures of quality. Many I/S organizations have failed to define and measure quality in their customers' terms. This failure is a failure to communicate. High costs are associated with the communication failure. Having failed previously to work with end users to define application quality in customer terms, I/S is often not included in the early stages of the line-of-business system planning activities of an enterprise and misses the opportunity for effective partnership with them. As a result, I/S often does not provide to the enterprise the full value of which it is capable.

In working with customers to define software quality, I/S has to recognize that quality includes both utility (usefulness) and usability. Utility refers to how much the application helps the user to complete his or her work. Usability is a combination of attributes such as user-friendliness and reliability. In order to be successful in meeting customer expectations, I/S development organizations must work with their customers to define quality for application development, and then must develop metrics that will help them manage to that definition of quality.

Ultimately, any definition of quality has to address both the products that I/S delivers and the software processes it uses to deliver them. These two broad dimensions of quality are often represented by the following questions:

1. Are we doing the right things?
2. Are we doing things the right way?

Table 1 illustrates the two dimensions.

Measurement drives quality. Quality experts are certain that measurement is essential to improving quality. According to Dr. Curtis Reimann, 1989 Chairman of the Board of Overseers of the Malcolm Baldrige National Quality Award, "The number one factor common to companies scoring high in quality was that they were quantitative and had instituted measurement processes." In other words, *measurement drives quality*. Well-designed measurement focuses on goals; reciprocally, well-defined quality is measurable.

Measurement is an essential component of total quality management. The old axioms are true: You cannot manage what you cannot measure; what is not tracked is not done.

Well-implemented measurement is an ongoing process of measuring work processes and products, finding out where the organization is compared to where it needs to be, and analyzing data to identify opportunities for improvement.[4] Measurement information is essentially *management information*. Well-designed measures identify the current capabilities of I/S, highlight opportunities for process improvement, facilitate goal setting, mark progress toward goal attainment, and enable benchmark comparisons with other organizations.

When used for process improvement, measurement can actually improve job satisfaction and morale because process improvements help people to work more effectively. The most successful measurement initiatives place a strong emphasis on employee involvement. Effective measures are *not* measurements *of* the people. They are measures of work processes made *by the people* and *for the people* so that process improvements can be defined and implemented. Effective measurement results in higher-quality products and in enhanced pride that people take in delivering these products to their customers.

The best metrics practices are linked to quality improvement efforts that harness the efforts and intelligence of everyone in the organization to find ways to do things better, from start to finish. The end result is lowered costs and increased productivity.

In application development, the cost of discovering and eliminating a defect increases dramatically as the application proceeds through the development life cycle. By moving defect detection efforts up to the early phases of application development, defect detection can be 33 times more cost-effective than testing done at the end of development.[5]

Hierarchy of measures. One of the most common problems that quality and metrics consultants encounter in I/S organizations is the absence of the concept of hierarchical information needs. Consider the case of the CIO (chief information officer) at a very large company, responsible for an I/S

When used for process improvement, measurement can actually improve job satisfaction and morale.

organization of 5000 people spread throughout several corporate divisions. The enterprise had already adopted TQM as its approach to improving its bottom line: market share and productivity. He knew appropriate measures were essential, so, being a business manager who had not come through the ranks of I/S, he asked I/S to propose the executive-level measures.

The I/S organization recommended that the executive-level measures be lines of code and person hours. These measures were ultimately rejected by the CIO because there was no way for him to relate such basic, low-level measures to the business objectives at his level. The measures told him nothing about how smoothly I/S development processes worked or whether the processes were under control, and they told him nothing about how satisfied I/S customers were, or how much I/S contributed to the performance of the customer organizations. What the CIO needed were the results of translating information up through the hierarchy of information needs of the organization.

Effective measures serve as a framework for defining shared goals and for communicating how the goals of each unit directly support organizational goals. Measures empower people at all levels by providing the data needed to make fact-based decisions. These data provide objective information about the working of the processes of the organization. Acting effectively upon the information requires management to shift the focus from personnel performance to process performance and from individual behavior to team behavior.

Successful approaches to establishing and using measurement recognize that each level of the organizational hierarchy has different information needs. The information needs will tend to be hierarchically structured: each level of the organization has to roll up information from its level into reports for the the next higher level. When measurement is designed to meet the hierarchy of information needs, the measures and metrics used tend to create a hierarchy of measures.

A hypothetical example may help to explain the concept of measurement hierarchies. Let us assume that the goal of one CIO is to increase responsiveness to the users of I/S by delivering applications faster. To do that, the organization has to shift more effort to development, away from maintenance. To support the CIO's goal of improved responsiveness, the I/S director establishes two goals: (1) increasing development quality and (2) decreasing maintenance costs. By achieving these goals, he or she will then be able to direct more of the total I/S effort to developing new applications.

For the application development managers, this goal translates to improving the quality of new applications delivered to users (I/S customers) and improving the maintainability of applications that are turned over to the maintenance group (the internal "customer" of the new-development group). Both new-development and maintenance project managers agree that reusing code is a good way to reduce error rates in new applications, thereby lowering the cost of delivering the application. They also agree that maintenance efforts should similarly benefit. Thus, they agree that quality and maintainability increases will initially be tracked by measures of code reuse and rework effort. The project managers decide to measure reuse and rework as follows:

- Percent code reused = Reused lines of code / Total lines of code
- Percent reusable code developed = Reusable lines of code developed / Total lines of code developed
- Percent rework = Total person hours devoted to rework / Total person hours of effort

Because each organization has its own particular culture, goals, and values, each organization needs to establish and internalize its own measures. This set of measures should be constructed to serve as critical business indicators, showing at a glance how the organization is doing.

Dimensions of I/S measurement

An essential step toward effective measurement is deciding what to measure, how to measure it, and how to use the measurement data. As discussed earlier, there are two broad dimensions of I/S quality: whether I/S is doing the right things, and whether it is doing them in the right way. Both of these dimensions need to be measured. This section presents two classes of I/S measures to be considered: business value measures and I/S process and product measures. Both classes of measures are needed to answer the question: Are we doing the right things in the right way? I/S process and product measures address the right way; business value measures address the right things.

Doing things right—I/S process and product measures. Doing things right means producing quality products and services as efficiently (which also means as cost-effectively) as possible.

I/S process and product measures provide insights into the capability of I/S to deliver quality applications. Process and product measures gauge process efficiency and outcome predictability by tracking such things as differences between estimates and actuals (e.g., effort, size, cost, maintenance requirements, and customer satisfaction). Information about project-specific attributes (e.g., project development platform and development methodology) is also required to appropriately group projects together when performing quantitative analysis.

Process and product measures are required to identify opportunities for meaningful improvement and to identify results (e.g., phase and cycle time, defects, rework) that significantly differ from the norm. The factors that contribute to favorable results can be propagated to other projects. Factors that contribute to undesirable results can be eliminated, thereby improving the software development and delivery processes.

Process and product measures are used to answer questions like "Are things getting better? Why? Are our improvement efforts effective?" Comparisons between pilot project results and historical norms, as well as overall trends, can test and demonstrate the effects of process changes.

The application developer's and tester's perspectives. Involving developers and testers in the measurement definition process will, in the long run, result in better estimates and better teamwork. Developers want measurable descriptions of the product to be developed, including customer expectations and quality requirements, in addition to project schedules and milestones. The developer is interested in measures that demonstrate (and support estimates of) how much effort is required to reach the target milestones and to deliver an error-free product. Providing straightforward ways to track current effort enables developers to be self-monitoring and self-correcting. When developers are self-correcting and when testers can see a record of the developers' testing activities, system testing costs will likely decrease. The tester wants to know how well the development group has inspected for errors, how many errors have been captured and corrected, and how large and complex the software is. Testers hope that quality will be built in, because they know how costly it is to try to "test it in."

The project manager's perspective. The project manager is expected to keep the project on plan. The project manager's measures need to track the project closely enough to provide enough lead time to anticipate problems and make timely course corrections. The project manager also needs to make and revise the project plan based on new information or changing conditions. Project estimates and revisions require accurate historical data in order to make reasonable budget and schedule projections.

Common project management measurement requirements include regular reports of progress (e.g., task starts and completions, milestone attainments), requirement changes that impact cost and scope (e.g., estimated versus actual impact of change requests), and defects (e.g., number of defects detected and defect severity broken out by phase).

The I/S development director's perspective. The I/S development director's perspective spans multiple projects. The development director needs measures that highlight meaningful differences between projects. The objective of the I/S development director's measures is to provide information for optimizing the software development processes of the organization. As owner of the development function, the I/S development director needs measures to identify factors contributing to or inhibiting quality and productivity, measures to evaluate the impacts of new tools and techniques, and measures to provide facts upon which resource allocation decisions can be made.

Business value measures—doing the right things. I/S products and services derive business value from improving the business performance of the enterprise by working with the lines of business to document, understand, and streamline their processes. Business value, then, is derived from *measured enhancement in business performance*. The amount of change can be measured by comparing initial performance to performance enhanced by I/S.

Involving developers and testers in the measurement definition process will result in better estimates and better teamwork.

Business performance is defined differently across different enterprises, depending on values specific to their defined missions. In order to maximize its business performance enhancement potential, I/S needs to be aware of, and plan according to, the values or strategic priorities of the planners of the enterprise. Table 2 presents commonly recognized categories of business value.[6] Each enterprise will place a different emphasis on different categories. Within each enterprise, each line of business may similarly emphasize categories differently, depending on current and future business needs.

The CIO's perspective. The CIO is in a uniquely advantageous position, bridging senior management planning activities that define the strategic directions of the enterprise and the automation and process re-engineering expertise of I/S. The stronger the bridge between enterprise planning

Table 2 Business value categories

Business Value Category	Category Description
Return on investment and wider economic impact	Business managers measure how much information technology leads to enhanced performance effectiveness of functional operations. It is important to know, for example, that recent enhancement of the billing system reduces billing errors by 70 percent, saving up to 35 hours a week (or $45,000 a year) in administrative reprocessing effort, and at the same time capturing up to 87 percent of revenues previously lost or deferred to the next billing cycle.
Management information	Supporting the ability of management to make the right decisions about core business activities and critical success factors is an increasingly highly valued activity. Today's decision-support systems help people throughout the enterprise, from enabling finance to give uniform, consistent, and timely reports and projections of business results, to helping to support sales and service by tracking the location and measuring the processing time for each service request and each customer proposal.
Competitive advantage	Competitive advantage springs from creating new *business opportunities*, improving the existing business position, or improving the way the industry or business is structured. Hertz's rental return agents meet returning drivers in the parking lot, immediately enter final trip data into palm-top computers, and in minutes the customer is off to catch a plane, final receipt in hand—no lines, no waiting.
Competitive response	Competitive response looks at the risk of *not* doing a project. It is a defensive action, concerned with preserving market share and with avoiding damage to the enterprise. For example, adding automatic teller facilities today may not help a bank gain market share, but it will help to keep up with the competition.
Strategic match	Strategically well-balanced I/S efforts promote success toward the strategic goals of the enterprise. They are consciously and deliberately initiated to *support corporate strategy*. The 1990 Baldrige winner, Federal Express, had the strategic goal of being the best and most reliable provider of overnight delivery services in the world. In support of this goal, the enterprise created information technology systems that enable anyone in the organization to know immediately the status of any of its 1 500 000 daily shipments.
Production capacity	Just as a state-of-the-art manufacturing facility is a significant asset to the business, so is its information technology, its I/S infrastructure and architectures. The existence of the manufacturing plant or of the I/S technology represents the existence of the *productive capacity* and production *flexibility* that enable the enterprise to compete effectively for future business. Without it, the company must find a substitute capacity (outsourcing) or endure the delay of building the required I/S capability.

and I/S planning, the more likely it is that I/S can maximize its contribution to the enterprise. Linking the two levels of planning facilitates the effective benefits assessment of proposed I/S efforts and promotes effective prioritization of I/S projects.

Thus, the CIO looks at the contributions that I/S can make to the enterprise as a whole and at the contributions it is currently making. When evaluating the contributions made by I/S application development efforts, the CIO requires timely, meaningful measures of customer satisfaction, I/S internal quality and productivity, and I/S business value contributions.

The CIO is also interested in *external benchmarking* to answer questions like:

- How do we stack up against other people? Against the best?
- If we are not the best, what can we learn from others who are?
- How does our experience with new technologies and methodologies compare with that of others?

Measures of business value provide the CIO with the data needed to understand the big picture: to know how much value I/S applications bring to its customer organizations, and to determine how much the value of I/S could increase through improvements in the capability of I/S to deliver quality applications. On the basis of current and potential business value, the CIO can make fact-based decisions when allocating the resources required for application development, application customization, maintenance, and process improvement.

Measuring I/S quality and value. The preceding subsections have looked at various factors that can be measured to assess whether I/S application development efforts are effective (focused on the right things) and whether application development is efficient (being done the right way). This subsection suggests a way to group I/S measures into measures of effectiveness and efficiency.

I/S effectiveness means focusing effort and resources where they will have the greatest positive business impact. Measurements of effectiveness usually include external perspectives:

- Customer satisfaction—Customer satisfaction can be measured by surveying user satisfaction with products and services supplied by I/S. Techniques to measure customer satisfaction may include customer surveys, evaluation questionnaires following up each delivery of a product or service, and customer forums, focus groups, or user groups. Customer surveys should include the business process owners, line-of-business executives, and application end users.
- Business impact—Business impact can be measured by evaluating I/S products and services in terms of their contributions to the competitiveness, productivity, and flexibility of the enterprise in daily business; that is, how much do I/S products enhance business performance? Assessing business impact includes follow-up on the "business cases" supplied by business units when requesting new I/S products or services, as well as pre- and post-install surveys. Business impact measures may include the business value of new markets or new revenue streams that an I/S product has enabled.
- Strategic alignment—Strategic alignment can be measured by evaluating the deployment of I/S products and services vis-a-vis the strategic goals of the enterprise. The alignment value of I/S is measured in relation to the alignment of I/S with business priorities. Strategic alignment may measure I/S effort expended per business unit in relation to the strategic value of that unit to the business. Alignment measures may also consider how closely I/S planning and enterprise planning activities are integrated.

Efficiency is the ratio of effective work to the energy expended in producing it. Efficiency measures the ability to produce a desired effect with a minimum of effort, expense, or waste. I/S measurements of efficiency are inward-focused; they relate to I/S internal activities and are typically technically oriented. Because they measure product against work effort, they consider:

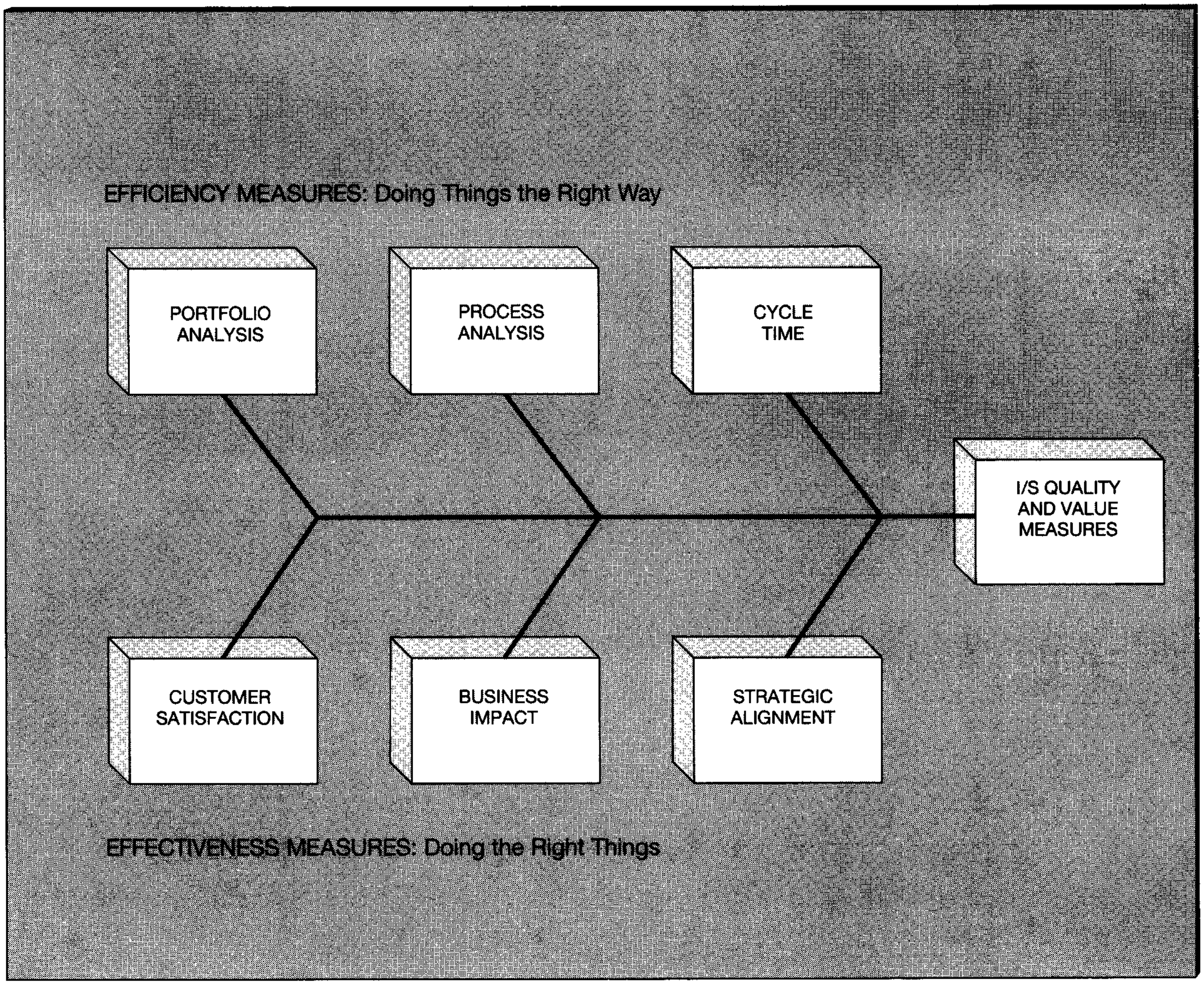
- Portfolio analysis—Portfolio measures may include the total number of applications, total size of the portfolio, portfolio cost per function point across lines of business, and annual trends in product quality or delivery cycle time. The measures should also include services offered, such as consulting, help desk, product documentation, and product training.
- Process analysis—Process analysis measures evaluate the repeatability and predictability of a process. They show whether or not a process is under control. The key here is how *dependable* the process is: Can budgets and schedules be estimated accurately? Can product quality be reliably predicted? Process analysis measures examine the rate at which work flows through each step of the process: the rate at which user requirements are translated into function delivered to the user. What percentage of overall development time is spent on a given phase of the development cycle, such as defining and validating requirements? How many errors are detected and corrected during each phase? How well are requirements traced through the process to delivery of the application?
- Cycle time—Cycle time measures responsiveness and shows the rate at which user requests for new products, enhancements, and services are fulfilled. Cycle time measures may include variances between projected and actual completion times, average and median times to deliver new applications and enhancements, and the rate at which backlog service requests are moved into active project status (a kind of "inventory turn" measure).

Figure 1 summarizes the factors that contribute to I/S quality and value.

Aligning process analysis measures with maturity

In the earliest years of software development, and in many organizations today, the goal of devel-

Figure 1 Contributors to I/S quality and value



opment was and is just to get the product out. Thirty years ago, there was no science of software engineering. People invented ways to build software at the same time as the software products were being built. Processes were *ad hoc*, and success often depended upon the "guru" leading the project. Fortunately, much has been learned over the past few decades.

A considerable body of knowledge has evolved, improving the ways in which software is designed, developed, and delivered. A new science, software engineering, has been born. Software engineering practices bring rigor and discipline to software organizations. Studies of commercial and governmental software organizations by the Software Engineering Institute (SEI) at Carnegie Mellon University have resulted in the definition of a software process Capability Maturity Model (CMM).[7] The process Capability Maturity Model distinguishes five levels of increasing maturity and capability.

Although the CMM does not provide an explicit formula for improving individual development organizations, it relies on empirical data that show a strong link between the maturity of the processes used within the software organizations and their ability to produce predictable results. Interestingly enough, the thesis that the ability of an organization to deliver high-quality products and services is linked to process maturity is also embodied in the criteria for the Malcolm Baldrige National Quality Award. The quality capability model of the Baldrige assessment also uses five levels of increasing capability.

Table 3 Quality models

Baldrige Score Range	Crosby's Quality Management (QM) Maturity Grid	SEI Capability Maturity Model
0% Anecdotal No systems evident	Uncertainty No comprehension of quality as management tool COPQ/Sales = 20%	Level 1—Initial No predictable results
10%–40% Beginnings of systematic prevention basis	Awakening Recognition of possible value of QM, but no commitment of resources COPQ/Sales = 18%	Level 2—Repeatable Basic project management processes Schedule predictability
50% Sound, systematic prevention basis that includes evaluation and improvement cycles; some evidence of integration	Enlightenment Management supports and helps QM COPQ/Sales = 12%	Level 3—Defined Cost predictability Standard development and maintenance processes as a basis for quality improvement
60–90% Sound, systematic prevention basis with evidence of refinements through evaluation, improvement cycling; good integration	Wisdom Management participates in and personally leads QM efforts COPQ/Sales = 8%	Level 4—Managed Formal methods, tools, technology use, and measurement Process measures; defect projections, control plans, and tracking; quantitative quality and performance goals
100% Sound, systematic prevention basis refined through evaluation, improvement cycling; excellent integration	Certainty QM considered essential part of corporate management systems COPQ/Sales = 2.5%	Level 5—Optimized Foundation for continuous improvement, optimization of the process Complete measurement and tracking systems

SEI's model was originally developed by Watts Humphrey, who described his approach to defining a framework for software process improvement as "roughly parallel" to a total quality management model mapped out by Philip Crosby.[8,9] Thus, it is not accidental that there is some correspondence between quality models and the Capability Maturity Model. Each suggests that the best way to improve software development quality and productivity is to improve the software development process. The five levels of the Baldrige Score Range, Crosby's Quality Management Maturity Grid, and the SEI Capability Maturity Model are summarized in Table 3. Crosby's Maturity Grid also shows the percentage of sales the cost of poor quality (COPQ) represents at each level of maturity.

Capability Maturity Model. Table 3 also illustrates the basic premise of the CMM: as software organizations mature, the processes they use mature. The CMM identifies five levels of software engineering process maturity ranging from Level 1, "initial" (the relatively chaotic situation that is still most common today), to Level 5, "optimized," the level of greatest maturity and highest quality.

At Level 1, the goal is simply to deliver software. There is no predictability for completion time, cost, quality, or functionality. Development processes are largely *ad hoc*. Usually, the only measures that are reliable are *post-ship* measures taken after the software product is delivered to the user. These measures often bring unpleasant surprises to management.

At Level 2 of software development maturity, the goal is project control. This goal is usually translated into schedule control. At this stage, projects are *schedule-driven*; cost control is often difficult because of the need to throw extra resources (such as large amounts of overtime) into the project so that a product is ready on time.[10] Unfortunately, often not enough hard historical data are available to enable reasonable schedule projections, so time begins to run out. In this "process," it is not unusual for functionality and quality requirements to fall by the wayside. As a result, costs to achieve customer satisfaction after the product is released can be very high, especially if there is a large installed base of users.

Level 3 works on *controlling the product* by assuring that all product requirements are met and quality-controlled. At Level 3, the expected product is produced on or close to schedule, but costs are still not under control. This situation is often attributable to the increased commitment to defect detection and removal. Because detection and removal efforts do not yet improve the process that introduces the defects, the cost of attaining quality is high.

At Level 4, the goal is to *control the process* by which the product is produced. Only by controlling the process can the costs of products, projects, and customer satisfaction be controlled. The expected product is delivered when predicted at the predicted cost. Now the organization can begin to improve processes and, by doing so, reduce cycle time, reduce costs, and produce better products.

At Level 5, the software organization builds on the foundation of control and improvement to establish a culture of continuous process improvement. Software development efforts focus on optimizing both quality and productivity. Although it is not a focus of the CMM, we can assume that the software organization itself has probably achieved much tighter integration with the business as a whole. As a result, measures that track the value to the business of the I/S organization as a whole are probably relatively mature.

Capability Maturity Model dominant measures. Table 4 shows samples of the measures currently advocated by the SEI (dominant measures) at each level of maturity.[11] Each change in focus for each level of maturity dictates a change in the measures used to manage software development. As the table illustrates, the dominant measures shift as the level of process maturity increases. This shift reflects three common axioms of measuring:

- What is not tracked is not done.
- You cannot manage what you cannot measure.
- What you *do* measure is what you get.

Let us look at those statements more closely. The information that measurement supplies is essentially a management information system (i.e., a system that supports decision-making). For that reason, the dominant measures relate to what the organization is trying to manage. As I/S organizations reach higher levels of software engineering maturity, the management information needs become more sophisticated. Processes are better designed, understood, and followed. Opportunities for "fine tuning the development engine" increase, driving the need for increased precision from existing measures and for collecting new measures.

The relationship between the I/S level of maturity and information needs is reflected in the table by changes in the dominant measures. As each succeeding level of maturity is reached, and additional measures are put in place, the dominant measures from the preceding level will occupy a smaller share of management interest and attention than it did previously.

The various sets of measures and metrics that the organization uses can be portrayed as gauges on an instrument panel. In arranging the instrument panel of the organization, managers might typically position the dominant measures as "continuous readout" dials (like speedometers and odometers). Other instrument panel areas would be assigned to "warning indicators" for measures and metrics that are of interest to management only when immediate action is required (like low-oil lights and door-ajar bells). As the organization reaches higher levels of maturity, new dominant measures will reconfigure the instrument panel: The new dominant measures will become continuous readout gauges and the dominant measures of the prior level will become warning indicators. Determining the appropriate I/S measures is therefore an ongoing process. Attaining a new level of software engineering process maturity or recognizing a shift in the enterprise business value model are indicators that it may be time to repeat the measurement selection process. Each time the measurement selection process is repeated, the effort required to define, collect, and analyze the measures should decrease.

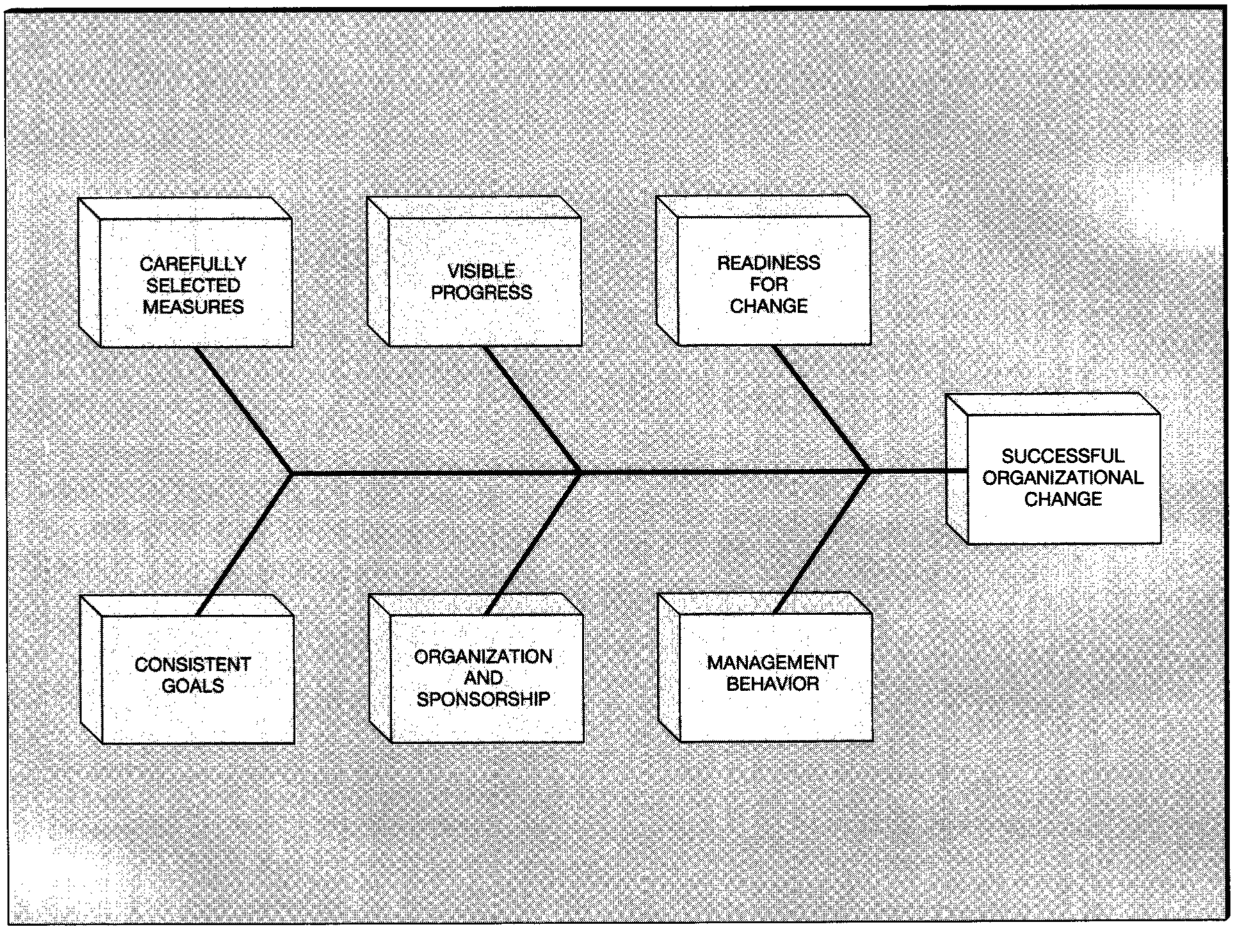
Table 4 CMM and dominant measures

CMM Level	Goal to Advance	Dominant Measures and Sample Measures
Level 1—Initial No predictable results	Achieve rudimentary schedule predictability	Post-ship measures • Customer-detected defects • Production failures • Application size at time of delivery • Total development cost • Total development time
Level 2—Repeatable Information or *ad hoc* process Rudimentary schedule predictability	Stable process; cost predictability	Project control • Trouble report density • Actual versus planned completions • Actual versus planned costs • Number of requirements changes • Application size growth
Level 3—Defined Cost predictability Stable process as a basis for quality improvement	Formal life cycle, methodology, tools, and technology	Product control • Defect detection efficiency • Defect frequency by defect type • Requirement change distribution by requirement type • Actual versus planned cost with ranges • Actual versus planned completion with ranges
Level 4—Managed Formal methods, tools, technology use, and measurement Process measures; defect control plans and tracking	Productivity plans, goals measures, tracking; process improvement; increased defect prevention	Process control • Progress in relation to control limits • Cost in relation to control limits • Size growth in relation to control limits • Defect removal efficiency • Rework measures
Level 5—Optimized Foundation for continuous improvement, optimization of the process Complete measurement and tracking systems	Continuous improvement of process(es); measurement system evaluation and refinement	Quality management • Actual versus predicted process improvement results • Reductions in process variance • Actual versus predicted process innovation effects • Innovation adoption rates • Rework effort as a percentage of total effort

Cultural and organizational dimensions of measurement

Successfully introducing measurement into an organization almost always requires significant cultural changes. Comprehensive planning for the cultural changes that must accompany I/S measurement is critical to the successful deployment of the process. Most people resist change. By understanding the dynamics of change, and carefully planning its introduction, I/S management can minimize stress for everyone involved. In fact, organizational change can be a positive, team-building experience. Successfully managing change requires coordination, communication, and commitment. Management must repeatedly reaffirm their commitment to the ultimate benefits gained from the change. These efforts are needed to reduce fear and stress. A strong communication effort should precede and constantly accompany the formal implementation of the measurement in order to minimize or forestall negative reactions people may have to the change.

Figure 2 Contributors to successful change



Organizational change is most successful when the entire organization:

- Recognizes the need to change
- Establishes and communicates the new shared vision
- Endorses the plan to arrive there
- Communicates, celebrates, and rewards progress

Cultural success factors for measurement. Optimizing the capacity of the organization for change is I/S management's challenge. Planning for successful change must address the culture of the organization as well as its structure. Appropriate leadership and sponsorship of the change are extremely important. Figure 2 illustrates the primary contributors to successful organizational change.

Carefully selected measures. The good news from measurement is: What you measure is what you get. The bad news from measurement is also: What you measure is what you get.

Regardless of how measures are used, the act of gathering measures focuses attention on what is being measured. People will naturally try to produce what is being measured, whether it be lines of code, fewer errors, or ease of use. Careful thought should go into selecting the measures that are right for a particular set of I/S information needs, objectives, and culture. For example, productivity targets stated only in terms of lines of code may cause developers to increase the number of lines of code (without adding any value to the application) and hence "improve" productivity as defined by the measurement plan.

Visible progress. The Quality Assurance Institute suggests that the plan should include milestones that mark short-term successes. Celebrating the attainment of these milestones reinforces the momentum behind the plan. In addition, focusing attention on positive results from the measurement process serves to affirm the commitment of management which, in turn, reinforces acceptance of the measurement process by the organization.

Readiness for change. Healthy organizational change begins when the organization, as a whole, recognizes the need to change. Many experts believe that three phases are involved: unfreezing, moving, and refreezing. Unfreezing is the process of recognizing the existence of a problem. Surveys or assessments (either self-assessments or external assessments) can assist the unfreezing process. Moving involves establishing a vision of the "brave new world," or the scenario after change takes place, as well as a plan to get there. Refreezing involves the implementation of the plan, monitoring progress, and making course corrections.

Consistent goals. The establishment of consistent goals from top business management to top I/S management to project management to the programmer is vital to the success of the program.

Top management goals such as return on investment or increased market share must be decomposed to specific measures for the project manager and the programmer. How the measures are combined to evaluate progress toward organizational goals should be clearly articulated and endorsed. In this way everyone can see how individual efforts relate to the big picture. For example, improved market share depends on improved quality, which depends on fewer defects, which depends on early defect identification, which depends on process improvement.

The real test of a measurement process is the degree to which everyone can make the translation from top management goals to the goals that each person is being asked to achieve. By passing this test, the measurement process helps the individual identify with the enterprise objectives and feel a part of them.

Organization and sponsorship. The owner of each process being measured must sponsor the deployment of its measurement. A team approach is highly recommended to determine the appropriate measures for a given process. Key people *who are participants in the process being measured* should be part of the planning and implementation team. Participants may also include measurement subject matter experts from quality assurance or the software engineering process group,[12] but in an advisory capacity. The key people on the deployment team should be the "informal" leaders in the organizations they represent. Their buy-in, or commitment to participate, is essential if the deployment is to succeed.

Management behavior. The managements of organizations that have successfully undergone organizational change have several traits in common. Among these are that management:

- View change as a process and prepare their organizations for evolution
- Understand that resistance is to be expected and develop plans to manage and minimize potential side effects of resistance
- Focus on team building and provide teamwork skill training and incentives for teamwork
- Create an environment where change comes from the "grass roots" but is directed toward a common vision

Assessing readiness for change. Change management experts have identified the following factors when assessing organizational readiness for change:[13]

- History of change—the prior experience of the organization in accepting change. Has a measurement process been attempted before? What happened?
- Clarity of expectations—the degree to which the expected results of change are shared across various levels of the organization. The classic example of conflicting expectations is the case in which a vague "quality" emphasis program is announced, perhaps even with a designated "quality executive," but the day-to-day management focus remains clearly fixed on meeting schedules, rather than on improving quality.
- Origin of the idea or problem—the degree to

which those most affected by the change initiated the idea or problem that the change solves. Packaged, "quick fix" measurement solutions often encounter "not invented here" problems. Similarly, measurement promoted by a department, external to the people involved in the process being measured, may not be well accepted. For example, developers frequently (and rightfully) resent interference from "watchdog" organizations that dictate what should be measured and why. This lack of acceptance is especially true when the people being asked to change do not believe there is a need to change.
- Support of top management—the degree to which top management sponsors the change. Top management support and involvement are evident when measures of the progress of the entire organization toward its goals are shared throughout the organization, when management requires that measurement results be included in all status reports, and when management authorizes allocation of sufficient time in the work week for people to produce, collect, analyze, and use measures.
- If top management pays only lip service to the change and does not provide leadership for change, if management fails to monitor the adoption of change and does not allocate the resources necessary to bring about the change, the rest of the organization will not take the change seriously. The change effort will sputter and die. Future cultural change efforts will face an even greater challenge.
- Compatibility with organizational goals—the degree to which the proposed change corresponds to past and present organizational practices and plans. For example, if management claims that "quality" is an important goal but only uses schedule-driven measurements, the organization will recognize the dichotomy and thwart the measurement process.

Concluding remarks

What to measure depends upon management information needs which, in turn, relate to enterprise-wide priorities. Senior I/S management wants to know if the I/S organization is doing the right things, in the right way. Both business value ("right thing") measures and I/S process and product ("right way") measures are required to satisfy the information needs of I/S management. The I/S measures should be organized hierarchically because of the hierarchical and interlocking nature of organizational, departmental, and project goals and information needs. Each level of the I/S organization must satisfy its own information needs *and* the information needs of the next higher level. The degree to which these information needs are satisfied is directly related to the degree to which the information needs can be quantifiably translated into a hierarchy of measures.

Although a picture of the full hierarchy may help communicate the ultimate vision of the measurement process, few organizations are capable of implementing the grand vision all at once. The ability to achieve an optimal hierarchy of measures and metrics ranging from application development projects up to and beyond the CIO depends on the following factors:

1. Complete understanding of the ability of the application development organization to measure[14]
2. Full alignment of business and quality goals up and down the organizational hierarchy
3. Selection and use of the most appropriate measures and metrics to track achievement of the aligned goals
4. Thorough assessment of the cultural changes required to integrate measurement into the way in which the organization carries out its business
5. Comprehensive planning for metrics implementation

The full value of a measurement process can be realized only if all of these factors are present and properly balanced.

Acknowledgments

The authors would like to acknowledge the following people for their patient review of and many helpful suggestions for this paper: Phil Braverman, Jim Constans, Jim Freeman, Maryanne Kuzniar, Bob McNeil, Steve Noble, Dennis Orton, Carolyn Schorr, Steve Thomas, and Barry Young.

Cited references and notes

1. *Critical Issues of Information Systems Management for 1991*, The Fourth Annual Survey of I/S Management Issues, Index Group (December 1990) and *Critical Issues of Information Systems Management for 1992*, The Fifth

Annual Survey of I/S Management Issues, Index Group, Cambridge, MA (December 1991).
2. L. Dobyns and C. Crawford-Mason, *Quality or Else*, Houghton Mifflin, Boston (1991), p. 168.
3. *Conference Proceedings, Quality Week 1991*, Software Research Inc., San Francisco (May 14–17, 1991).
4. This process is embodied in W. Edwards Deming's "Plan-Do-Check-Act" cycle and in Juran's continuous improvement model and is integral to H. James Harrington's business process improvement model.
5. G. W. Russell, "Experience with Inspection in Ultra-Large Scale Development," *IEEE Software* **8**, No. 1, 25–31 (January 1991).
6. M. M. Parker and R. J. Benson with H. E. Trainor, *Information Economics, Linking Business Performance to Information Technology*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1988).
7. M. C. Paulk, B. Curtis, M. B. Chrissis, et al., *Capability Maturity Model for Software, Version 1.1*, CMU/SEI-93-TR-24, Software Engineering Institute, Pittsburgh (January 1993).
8. W. S. Humphrey, *Managing the Software Process*, Addison-Wesley Publishing Co., Reading, MA (1989), p. 4.
9. P. Crosby, *Quality Is Free*, McGraw-Hill Book Company, New York (1979).
10. A schedule/cost/quality model developed by Quantitative Software Management, Inc., suggests that shortening the schedule by adding project team members may adversely affect quality.
11. J. H. Baumert and M. S. McWhinney, *Software Measures and the Capability Maturity Model*, CMU/SEI-92-TR-25, Software Engineering Institute, Pittsburgh (September 1992).
12. W. S. Humphrey, *op. cit.*, p. 31.
13. M. M. Danziel and S. C. Schoonover, *Changing Ways: A Practical Tool for Implementing Change Within Organizations*, AMACOM, a Division of American Management Association, New York (1988), pp. 15–16.
14. This paper suggests that the ability of the organization to measure is linked to its level of software engineering process maturity and that its "dominant measures" should be selected based upon the level of maturity of the organization.

General references

The following selections are suggested for those interested in learning more about software measurement, quality, and productivity.

Software measurement

A. J. Albrecht and J. E. Gaffney, "Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation," *IEEE Transactions on Software Engineering* **SE-9**, No. 6, 39–64 (November 1983).

B. W. Boehm, *Software Engineering Economics*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1981).

R. C. Camp, *Benchmarking—The Search for Industry Best Practices That Lead to Superior Performance*, ASQC Press, Milwaukee (1989).

C.-K. Cho, *Quality Programming: Developing and Testing Software with Statistical Quality Control*, John Wiley & Sons, Inc., New York (1987).

S. D. Conte, H. E. Dunsmore, and V. Y. Shen, *Software Engineering Metrics and Models*, Benjamin/Cummings Publishing Co., Menlo Park, CA (1986).

B. J. Dreger, *Function Point Analysis*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1982).

A. S. Duncan, "Software Development Productivity Tools and Metrics," *Proceedings of the 10th International Conference on Software Engineering*, Singapore (April 11–15, 1988), pp. 41–48.

R. B. Grady, *Practical Software Metrics for Project Management and Process Improvement*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1992).

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

W. S. Humphrey, *Managing the Software Process*, Addison-Wesley Publishing Co., Reading, MA (1989).

IEEE Software Productivity Measurement Committee, *Draft Standard*, IEEE Computer Society (1990).

C. Jones, *Applied Software Measurement*, McGraw-Hill Book Co., Inc., New York (1991).

T. McCabe, "A Complexity Measure," *IEEE Transactions on Software Engineering* **SE-2**, No. 4, 308–320 (December 1976).

H. D. Mills and J. H. Moore, "Bringing Software under Statistical Quality Control," *Quality Progress* **XXI**, No. 11, 52–55, ASQC Press, Milwaukee, WI (November 1988).

J. Musa, A. Iannino, and K. Okumoto, *Software Reliability: Measurement, Prediction, Application*, McGraw-Hill Book Co., Inc., New York (1987).

H. M. Parsons, "What Happened at Hawthorne?", *Science*, No. 183, 922–932 (March 8, 1974).

L. F. Urwick and E. F. L. Brech, *The Hawthorne Investigations, Volume III: The Making of Scientific Management*, Management Publications Trust, London (1949).

R. Werling, *Data Collection System for Estimating Development Cost*, U.S. Air Force Contract No. F33615-85-C-5123, Defense Technical Information Center, Alexandria, VA (30 SEP 86).

H.-T. Yeh, *Software Process Quality*, McGraw-Hill Book Company, New York (1993).

H. Zuse, *Software Complexity: Measures and Methods*, Walter de Gruyter, Inc., Hawthorne, NY (1990).

Quality

R. Aguayo, *Dr. Deming*, Simon & Schuster, New York (1990).

K. R. Bhote, "Motorola's Long March to the Malcolm Baldrige National Quality Award," *National Productivity Review* **8**, No. 4, 365–375 (Autumn 1989).

C. W. Burrill and L. W. Ellsworth, *Quality Data Processing—Profit Potential for the 1980s*, Burrill-Ellsworth Associates, Inc., Tenafly, NJ (1982).

W. E. Deming, *Out of the Crisis*, Massachusetts Institute of Technology, Center for Advanced Engineering, Cambridge, MA (1986).

W. E. Deming, *Quality, Productivity and Competitive Position*, Massachusetts Institute of Technology, Center for Advanced Engineering, Cambridge, MA (1982).

J. F. Halpin, *Zero Defects—A New Dimension in Quality Assurance*, McGraw-Hill Book Co., Inc., New York (1987).

H. J. Harrington, *Business Process Improvement—The*

Breakthrough Strategy for Total Quality, Productivity, and Competitiveness, ASQC Quality Press, Milwaukee (1991).

H. J. Harrington, *Excellence—The IBM Way*, ASQC Quality Press, Milwaukee (1988).

M. Imai, *Kaizen—The Key to Japan's Competitive Success*, Random House, New York (1986).

C. L. Jones, "A Process-Integrated Approach to Defect Prevention," *IBM Systems Journal* **24,** No. 2, 150–167 (1985).

G. G. Schulmeyer, *Zero-Defect Software*, McGraw-Hill Book Co., Inc., New York (1990).

Software Design Inspections, Software Productivity Research, Burlington, MA (1990).

User Satisfaction Survey Questionnaire, Software Productivity Research, Burlington, MA (1989).

1990 Application Guidelines: Malcolm Baldrige National Quality Award, U.S. Department of Commerce, National Institute of Science and Technology, Gaithersburg, MD (1990).

Productivity

T. K. Abdel-Hamid, "Investigating the Cost/Schedule Trade-Off in Software Development," *IEEE Software* **7**, No. 1, 97–105 (January 1990).

B. W. Boehm, "Improving Software Productivity," *Computer* **20**, No. 9, 43–57 (September 1987).

B. W. Boehm, "A Spiral Method of Software Development and Enhancement," *Computer* **21**, No. 5, 61–72 (September 1988).

F. Brooks, "No Silver Bullet: Essence and Accidents of Software Engineering," *Computer* **20**, No. 4, 10–19 (April 1987).

R. Carlyle, "Leaping Ahead in Software Productivity," *Datamation* **35**, No. 23, 22–32 (December 1, 1989).

W. B. Chew, "No-Nonsense Guide to Measuring Productivity," *Harvard Business Review* **66**, No. 1, 110–118 (January-February 1988).

L. Cohen, "Quality Function Deployment: An Application Perspective from Digital Equipment Corporation," *National Productivity Review*, 197–207 (Summer 1988).

T. DeMarco and T. Lister, *Peopleware: Productive Projects and Teams*, Dorset House Publishing, New York (1987).

C. Jones, *Programming Productivity*, McGraw-Hill Book Co., Inc., New York (1986).

B. King, *Better Designs in Half the Time: Implementing Quality Function Deployment in America*, GOAL/QPC, Methuan, MA (1987).

J. Martin, "Gain Management Support by Measuring Productivity," *PC Week* (January 8, 1990), p. 76.

J. Martin, "Integrated CASE Tools a Must for High Speed Development," *PC Week* (January 22, 1990), p. 78.

R. J. Norman and J. F. Nunamaker, Jr., "Case Productivity Perceptions of Software Engineering Professionals," *Communications of the ACM* **32**, No. 9, 1102–1108 (September 1989).

Accepted for publication March 15, 1993.

Charlene ("Chuck") Walrad *12 Brooke Circle, Mill Valley, California 94941-4604.* Ms. Walrad is a software management consultant specializing in cross-life-cycle quality and productivity and their measurement. She has instituted programs and techniques to introduce total quality management within a variety of commercial software companies and I/S organizations. She has firsthand knowledge and experience of the corporate-wide quality programs at Xerox, Intel, and IBM. She has implemented a wide range of software quality and productivity processes, spanning all phases of the software life cycle and promoting cross-functional teamwork. These processes include design and development inspections and reviews, phase gating, formal configuration management, test automation, measurement and tracking programs to identify and eliminate sources of customer dissatisfaction and to reduce support costs, and quality audits of third-party software vendors. Prior to becoming a consultant in 1987, Ms. Walrad managed the design, production, or support, or both of the latter, of more than two dozen commercial and government software products, including mission-critical systems for U.S. Intelligence and the Commission of the European Communities. She has had budget and schedule responsibilities for multimillion-dollar software projects as well as small projects, has managed small teams as well as large organizations, and has managed remote operations in Europe, Mexico, and Japan. Ms. Walrad has been an invited speaker at conferences, at the U.S. Department of Defense, the Department of Commerce, the Central Intelligence Agency, and the National Academy of Sciences. She has taught at the university level, coauthored a college textbook, been selected for membership in Mensa, and is included in *Who's Who in the West* and *Who's Who of Global Business Leaders.*

Eric Moss *IBM Consulting Group, 1111 Broadway, 17th Floor, Oakland, California 94607.* Mr. Moss is part of the IBM Consulting Group and is a member of a team developing IBM's consulting methodology for application development quality, productivity, and measurement services. He has consulted with several IBM internal I/S organizations to implement function point analysis. Prior to joining IBM, he developed integrated financial modeling systems for several major public utilities, authored a planning policies and procedures manual for a major West Coast utility, and developed statistical sales forecasting models for a major consumer package goods company. Mr. Moss holds a Bachelor of Science degree in management science from Carnegie Mellon University and a Master of Business Administration from the University of California, Los Angeles. He has also been recognized as a Certified Quality Analyst by the Quality Assurance Institute.

Reprint Order No. G321-5520.