Managing Quality and Quality Improvement

David C. Elmendorf

In 1986, the managers for AT&T's packet-switching services determined that development costs, development schedules, and number of defects needed to be improved to increase the competitiveness of these services. In response to the demands of the service managers for improvements, the Packet Switching Development Department began to work aggressively on all aspects of product quality. Over the ensuing five years, the packet-switching development team (i.e., service planning, systems engineering, and development) applied the principles of *project quality management and improvement* (PQMI) to all aspects of the development program. The results have been impressive. Product defects discovered after delivery of the product have decreased 80 percent. The cost structure has decreased more than 40 percent, and cycle time for the development of AT&T Accunet® packet services has decreased more than 50 percent. This paper presents highlights of the application of the PQMI principles to a real-life quality crisis. It emphasizes the role of management and customer focus in developing the quality-improvement program.

Introduction

The quality-management program described here for AT&T's packet-switching development project began in response to the needs of an internal customer, i.e., the organization that provides AT&T Accunet packet-switching services. This customer—Data Communications Services—is part of the AT&T Communications Services Group, and its mission is to provide data communications services to external customers, i.e., to users of AT&T Accunet services. (The term *internal customers* refers to our colleagues and other internal organizations that depend on the output or results of our work. The term *external customers* refers to the people or businesses who purchase the company's products and services.)

Before this quality-improvement program was implemented, the project's customer was unable to accomplish its business mission because of the quality and cost of the products it received. The managers for AT&T's packet-switching services felt that the number of product defects was too high, development costs were too high, and development schedules were too long. They demanded improvements.

At the customer's urging, the project team (which consisted of service planning, system engineering, and development) developed and put into place a customer-focused set of changes to the project's approach to quality. (Panel 1 defines acronyms and terms.)

The Key Premises. Our program for quality improvement has three key premises:

- *Quality is measured by customer delight.* It is impossible to talk customers into being delighted. We cannot explain to them why they really should be delighted. *Delight* is the customer's sense that all its business requirements and expectations are being met or exceeded. Thus, customer delight became the *goal* of the quality program.
- *Quality includes product performance (i.e., the lack of product defects), product and support costs, and product-delivery interval.* All three are necessary if we are to meet the

Panel 1. Abbreviations, Acronyms, and Terms

customer-supplier model — a work process that emphasizes relationships between the customer and supplier, process input and output, and requirements and feedback

customer delight — customer's sense that all its business requirements and expectations of a product or service are being met or exceeded

external customer — a person or business that purchases the company's products and services

internal customer — a colleague or organization that depends on the output or results of our work

management process — an activity that establishes responsibilities, defines the process, manages relationships between the customer and suppliers, evaluates process performance, and identifies opportunities for improvement

methodology — the processes, metrics, and documentation developed for a particular task or technique

metrics — standard measurements that are used to compare the results to the requirements

PQMI — process quality management improvement; a seven-step methodology for process management and continuous improvement

process improvement — activities that introduce beneficial change to a process

product realization process — a design, development, and manufacturing stage required to take a product or service from its initial concept to its delivery to customers

Tier IV support — final escalation level for resolving customer problems

business requirements of the customer. Hence, performance, costs, and delivery interval define the *scope* of the quality program. Delighting the customer with all three concurrently requires that significant effort be placed on management of the project and its deliverables.

- *Managers are the key to the customer's perception of quality.* The manager guides the organization and sets its goals. Thus, he or she must be the quality program's *key enabler.* If the manager's guidance does not include the insight and leadership required to understand and delight the customer, no verification or rework steps can make up for it.

The Results. As a result of the program we built that is based on these premises, our internal customer became delighted with the project's performance. Costs were being controlled, commitments were being met, and the product was meeting the needs of internal and external customers the first time.

On many occasions, the project team has had to return to these basic principles as problems developed. We have always found that new quality problems were rooted in our losing sight of the goal, the scope, or the key enabler.

Driving Forces Behind the Program

Starting a quality program requires considerable energy. Quality improvement implies change, and change—for whatever reason—is not readily accepted. The project's customer (i.e., Data Communications Services) and the project's management were the driving forces behind this quality program.

Our customer had its business goals to achieve. But without predictable quality, that was not possible. Various upper-management directives were holding the project's managers and our customer's managers accountable for leading their organizations in a way that would ensure they produced quality products.

This combination of forces was required to provide the impetus to overcome organizational inertia.

The Customer as a Driving Force. If we were to use the customer's input as a driving force, then the project's management had to learn to listen. We needed to treat the customer's input as a treasure.

We found that it was difficult to receive the quality message from our customer. We wanted to believe we were creating a quality product, and we became defensive. Our customer only knew that its needs were not being met and became demanding.

To lead the way to resolving this impasse, the project managers insisted that all project members set aside their defensiveness and listen to the customer's view. It was critical that the individuals who managed the project provided leadership by insisting that we put the customer's view of quality first.

The Management as a Driving Force. Two realities—that we were not managing for quality, and that management was part of the problem—were also

difficult to accept. There were perceptions among the technical staff that quality was not important and was not rewarded by the organization.

To alter this perception drastically required management's personal daily involvement and the right emphasis in the performance-review process. We had to learn how to reward people for participating in the process of quality improvement. We also had to learn to avoid the trap of using quality metrics as the basis for rewarding people for quality.

Basic Principles of Quality Management

With the driving forces in place, the change process could begin to work. All approaches to quality improvement have their roots in the Shewhardt cycle.[1] This cycle is a model for continuous quality improvement that was developed in 1939 by W. A. Shewhardt and was made famous by W. E. Deming through its use in his work in Japan in the 1950s.

Figure 1 shows the version of the cycle that we applied to the project to enable us to structure our quality program. It graphically depicts AT&T's formalized methodology for dealing with process quality management and improvement.[2] This seven-step method, which is known as PQMI, gives AT&T managers the means to establish process management responsibilities by organizing a team of accountable individuals to define process tasks. Often, PQMI is coupled with techniques for project management and continuous quality improvement.

The PQMI approach was simple to understand, but its application to a complex development process required considerable planning and mangement. Without careful planning and management, changes that were intended to improve quality decreased quality instead. The methods we developed for applying the cycle to the project are described in this section. We also explain specific methods and important lessons associated with the application of each cycle step of the PQMI approach.

Step 1—Identify Process Owner. Before change will take place, responsibility for the quality of a function or process has to be established. The first step in identifying ownership of a process is to decide what the project's processes are.

We determined that there are two kinds of processes in an organization: product realization and management. *Product realization processes* are those activities that create the organization's deliverables to its customer. These processes represent the design, development, and manufacturing stages required to take a product or service from its initial concept to its delivery to customers. *Management processes* are those activities that allocate and control the use of resources. These activities establish responsibilities, define the process, manage relationships between the customer and suppliers, evaluate process performance, and identify opportunities for improvement.

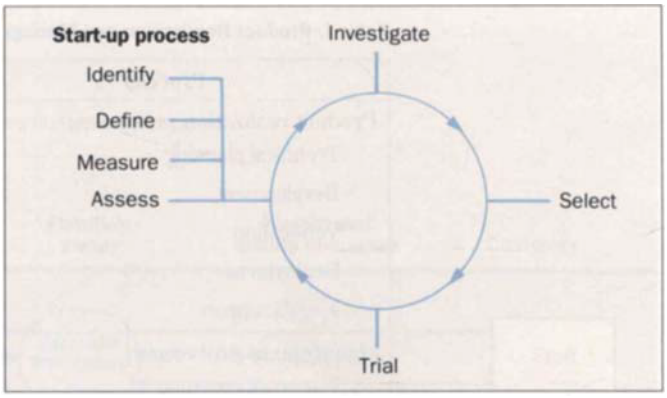

Figure 1. The process quality management improvement (PQMI) model used by the packet-switching development project. This version of the Shewhardt cycle emphasizes the start-up process steps (i.e., identify, define, measure, and assess), which must be completed before the rest of the cycle can begin.

Table I identifies the project's processes and goals in each category. If each participant in the project is to do his or her job with quality, then the technical staff, management, and customers need clear, agreed-on statements of the processes and associated goals.

A breakthrough for the project's management was to treat the management processes as distinct and independent from the product realization process. By keeping the processes separate, management was able to focus on how well it was doing its job rather than on attributing all quality problems to the technical staff.

Process ownership clarifies the ownership of execution quality. Staff members can only be responsible for the quality of those processes for which they have adequate managerial support. If staff members fail to deliver a quality product even though managers have provided adequate support, then the problem becomes one of staff execution. That is, the staff

Table I. Product Realization and Management Processes

Process	Goal
Product realization processes:	
Technical planning	Design architecture for product or service
Development	Design, implement, and test a fit-for-use product
Verification	Verify that the product meets customer expectations
Deployment	Distribute technology into service context
Tier IV support	Resolve escalated customer problems
Management processes:	
Plan work program	Define performance, cost, and interval commitments that meet the customer's business requirements
Organize resources	Structure and assign projects within the organization to meet each commitment
Manage process	Monitor and adjust project execution to achieve quality commitment
Evaluate quality and improve process	Determine gaps in processes' ability to delight the customer; propose improvement

members own the execution problem.

Management is responsible for providing the support that staff members require. However, that support includes resource allocation, process definition, and cross-organizational escalation (i.e., allowing a person to go outside his or her organization to find the solution to a problem). If managers fail to recognize and execute these activities as required, then the problem becomes one of management execution.

Using process ownership. When we began to distinguish between staff and management execution, complex quality issues became relatively simple to address once we resolved ownership of the problem.

For example, two quality problems our customer felt were important were *defects not corrected* and *defects introduced* during the generation of software repair packages. By viewing these problems as potentially either management execution problems or staff execution problems, we were able to broaden our search for the source of the problem.

We found that a process the management had established had inadequate mechanisms for:

- Monitoring problems with the repair packages
- Stopping delivery of defective repair packages
- Requesting rework.

As a result, the staff members could not produce good quality without going outside the defined process by requesting rework themselves from their friends. As long as an experienced staff member executed the process, problems did not occur because he or she always bypassed the existing process. But when a new staff member executed the process, failures occurred because he or she used the defective process—which did not provide for rework.

Thus, management had created a defective process that could not produce good quality. Ownership of the problem belonged to the managers who owned the process. By recognizing that fact, we were able to fix both the management process that allowed a faulty process to exist and the specific process of repair-package generation. This change improved not only the process where the faults were being found, but also the quality produced by many processes.

Management process as the problem. Poor management processes alone accounted for an important problem with product delivery in 1989. In our internal customer's view, a particular delivery was four months late. There were two reasons for the problem:

- During the planning phase of the project, an informal agreement had been reached to provide a "friendly user" interim delivery to the customer. However, management had not formalized this agreement. Later, the customer translated the agreement into delivery of a *fit-for-use* product.

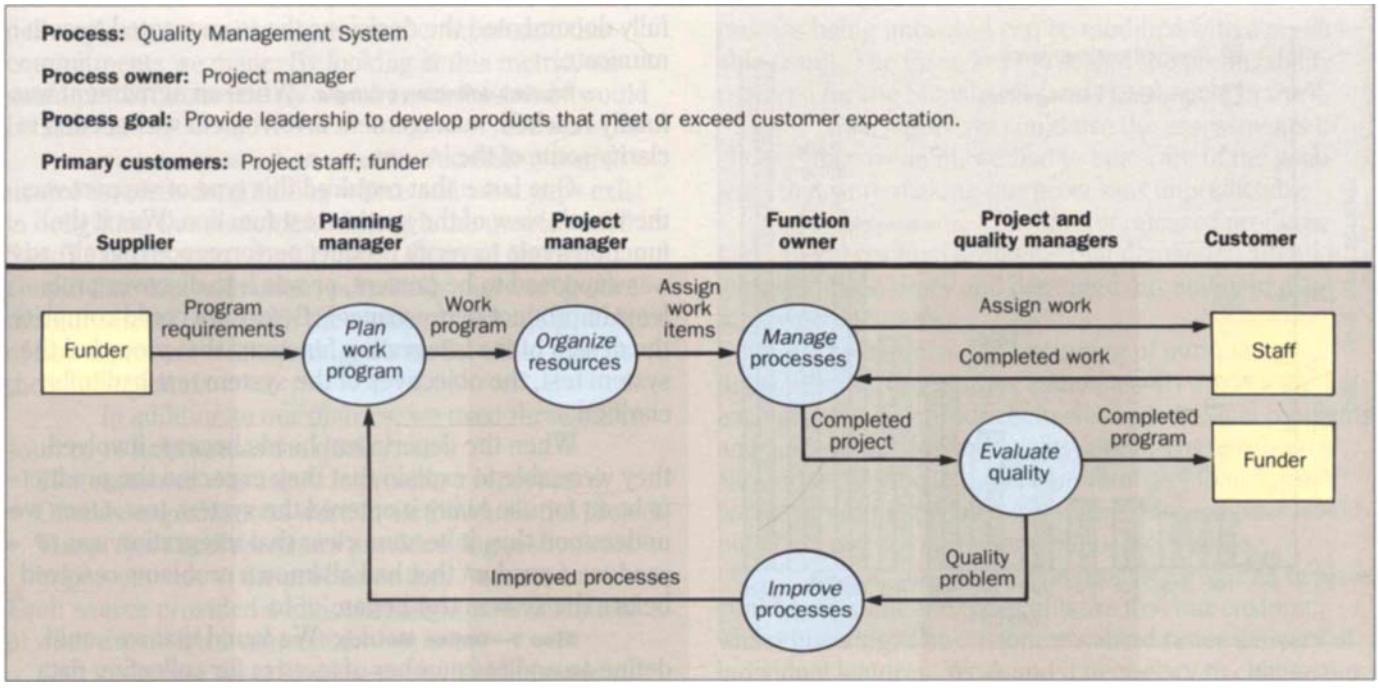

Figure 2. A graphical presentation of a process that clearly states the owner, goal, customer, suppliers, and internal deliverables is a critical communications and teaching aid. Feedback paths are all assumed, as required by the customer-supplier model.

- During the monitoring phase of the project, supervision had not responded to alerts from the technical staff that the quality of certain products from external vendors was not adequate.

Together, these problems with the management process created a serious quality failure from the customer's point of view. The solutions resided 100 percent in the management processes.

Step 2—Define Process. To design a process, we used a structural decomposition procedure that is similar to the *top-down design* of software or hardware. First, we established boundaries for each process by defining its deliverables. We then decomposed the process into the functions and relationships required to create those deliverables.

This seemingly simple activity of naming and providing boundaries for our project's processes took several months for two reasons:

- Management's limited understanding of how the processes really worked
- Substantive disagreements about the objectives of certain functions.

The design of each process required the active participation of three levels of staff. Each level provided a different perspective and responsibility to process design. The *project manager* provided leadership for the cultural change and enforcement of commitment to reach closure. That is, he or she had to get all participants to agree to the decisions and commit to changes. The *supervisors* provided commitment to the customer and supplier agreements about deliverables. The *execution level* provided understanding about the deliverables and what it takes to get the job done. If we remove any of the three levels from the design process, then the inputs required for complete agreement will be missed.

A mechanism we found useful to achieve closure was the design-review process at the management level. The review was necessary to identify and clarify agreements and to achieve project *buy-in* (i.e., commitment and ownership). This review followed the same rules as a code inspection—with a moderator, defect list, and problem-resolution process. The stakeholders in the processes under review served as the inspectors.

Process-definition communication tool. A process

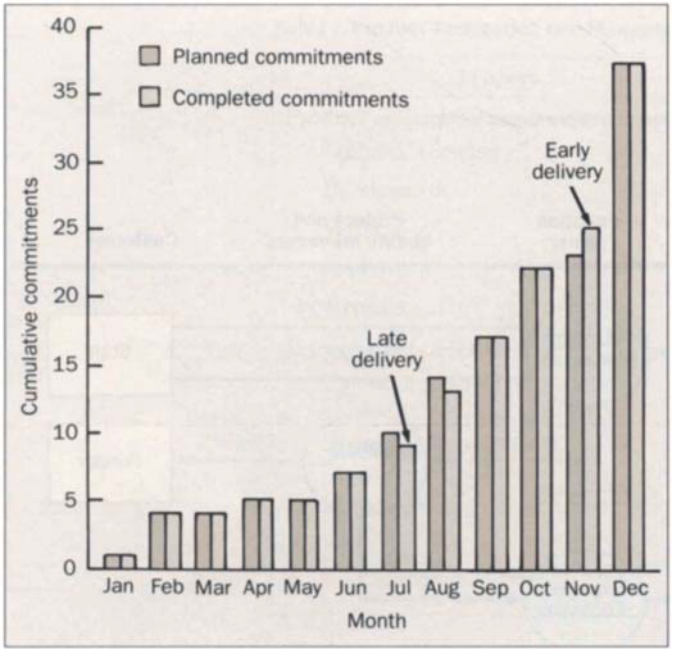

Figure 3. Commitment tracking for the packet-switching project. Commitments met is a simple measure of a process quality that is important to customers. This histogram shows how many of the delivery-date commitments had been met in 1990.

design has the following elements: the owner, deliverable, customer, supplier, and relationships between activities. Figure 2 gives an example of a process definition as expressed in a style the project's management developed.

This graphical, one-page format became an important means of communication. By using this format, we found we could quickly identify areas of disagreement and focus our discussion on them. The development of the diagram for a particular process required the participation of all customers, suppliers, and step executers—along with the process owner. When completed, the diagram also became a useful teaching tool for communicating process-design decisions to new project members.

We developed this tool in response to the frustration of project members over our collective inability to reach agreements on process designs even after endless meetings. The graphical depiction was selected and modeled after techniques used in software design methodologies. The specific depiction template we used fully documented the decisions the team wanted to communicate.

Process-definition example. When an agreement was finally reached, management involvement was needed to clarify some of the issues.

One issue that required this type of support was the team's view of the system-test function. Was it the function's role to verify product performance that already was supposed to be present, or was it to discover problems in product performance? Before we could complete the design of the integration functions that preceded the system test, the objectives of the system test had to be clarified.

When the department heads became involved, they were able to explain that they expected the product to be fit for use *before* it entered the system test. Once we understood this, it became clear that integration was to produce a product that had all known problems resolved before the system test began.

Step 3—Define Metrics. We found that we could define an endless number of metrics for collecting data on a process, because we confused descriptive and diagnostic quality measurements. *Descriptive measurements* tell us whether a process is working, while *diagnostic measurements* tell us what is wrong with the way a process is working.

Although diagnostic measurements were important to have, they did not necessarily tell us if a customer would be delighted with the results of the process. Our shift to descriptive measurements gave numeric evidence that the customer would be delighted.

Selection of metrics. Earlier in the history of the project, we had spent a lot of time looking at the faults found per thousand lines of software after delivery to system test. We were able to bring down the fault count consistently, but did it *without* delighting our customer. The measure of quality we had focused on was important to us, but was not important to our customer.

Therefore, the project team selected customer-focused metrics for each deliverable from each process. The metrics were in the categories of defects delivered, cost, and the development time needed (i.e., successful prediction of the delivery date). These categories ultimately represented the customer's concerns.

A useful metric. Figure 3 shows one customer-focused metric that we chose to measure ourselves the way our customer sees us. That is, it shows how many

delivery-date commitments we met compared to the commitments we made. By looking at this metric, we could predict immediately whether our customer would be satisfied.

Step 4—Assess Conformance. Assessing conformance involves determining where and what gaps exist in delighting the customer. Although the metrics used in Step 3 gave us some conformance data, we needed to couple that data with other, potentially more subjective measurements. Although the numbers gave us a model of how a process was working, no model could give us the whole picture.

In addition to our metrics, we used three other sources to help assess the quality.

- The customer was asked to provide a report card.
- Outside organizations were invited to audit the project.
- The project staff members provided suggestions for improvements to the processes they executed.

Each source provided additional insight and validation of what the metrics suggested.

Disciplined use of assessment data. End-of-cycle assessment had to become a regular part of our development process. However, this assessment was an activity that few people were motivated to do. If the development succeeded, they wanted to get on with the next development. If it was not so successful, they were sensitive to calling attention to their lack of success. Thus, it became the role of management to ensure that the assessment was done and lessons were learned for each cycle.

The assessment's result was a statement of the gaps in our processes that kept the project from delighting the customer. The management team analyzed the gaps and then selected processes for improvement, based on the result expected.

Steps 5 through 7—Improve Process. Up to this point, the steps had merely laid the groundwork for quality improvement to begin.

Before the groundwork just described was in place, the improvements we tried had merely moved problems from one part of the process to another. We found that three more keys were needed to be successful when making improvements:

- Stable products
- Properly sized tasks
- Capable and empowered people.

The Shewhardt cycle (and, therefore, the PQMI cycle) is based on the underlying assumption that the process being improved can be modified with a predictable result. The three keys provided the predictability required for the Shewhardt (and PQMI) cycle to work.

Thus, before we could use the assessments to choose improvement, we had to take care of the problems that were making our processes unpredictable.

Stable products. Our base of released products had several residual problems that demanded attention at inopportune times and disrupted our ability to plan new work accurately.

Therefore, at the beginning of our quality-improvement program, we issued a software release for each of our major products to clean up residual problems and achieve the desired quality levels. These releases allowed us to proceed with commitments to new work and assured that problems with the released base would not affect our commitments to new work items.

Properly sized tasks. The processes we had in place could not handle the types of tasks that our customer was requesting. The customer wanted faster delivery of individual features. We found it necessary to change our processes and tune the processes to the nature of the customer's request.

This solution meant we had to establish a process that would handle several small commitments, rather than a few large ones. The project moved from handling 2 to 4 large tasks in a year to handling 30 to 40 very small ones.

Capable and empowered people. The existing organizational structure did not give people the incentive for change. We had a centralized, quality-management structure, which meant that a quality-assurance team was responsible for planning and implementing all quality improvements. Thus, the process owners did not have any incentive to make their processes work better. Instead, the organizational structure provided needless opportunities for conflict between the process owners and the quality manager.

Empowerment came about when we made the process owners responsible for developing their own improvement program.

First, the quality manager developed the process for improvement and suggested areas for improvement, based on the project-wide deficiencies that the project audits had identified. Then, the process owners were responsible for designing and implementing the proposed improvements. This made it possible for the

Table II. Improvements to Plan-Work-Program Process

Improvement	Effect on customer-focused metrics
Smaller deliverables planned independently	Development interval decreased more than 50 percent
Customer expectations documented at time of project commitment	Serious defects found after product introduction decreased by more than 80 percent
Project processes and internal deliverables used to estimate projects	Budget requested for nondesign work (i.e., loading) decreased more than 40 percent
Project estimates used to allocate and track resource usage	Cost agreements consistently achieved

project to handle more changes and gave the pride of accomplishment back to the process owners.

Example of Quality Improvement

The process *plan work program* (defined in Table I) was a problem area because, in our customer's assessment, the project planners were requesting unacceptably large development intervals and budgets to do the jobs the customer needed done. Our investigations determined that the project planners had been using historical intervals and budget assumptions to lay out the work program. These intervals were traditional two-year cycles, and the costs were based on building and maintaining a "luxury" version of the system. Our customer was also dissatisfied because of the low correlation between the planned cost and schedule and the actual cost and delivery date.

As a result of these quality problems, our customer was unable to make the effective predictions necessary to run its business.

Predictable Within Specification. The goal of the plan-work-program process is to predict for a project deliverable the performance, cost, and delivery date that meet the customer's business needs. With few exceptions, the packet-switching development project did not have the ability before 1986 to predict these elements accurately.

Continuous improvement that is based on customer-focused metrics is required to delight the customer. Table II gives the specific process improvements that were selected and tested. It took a few years to effect all the changes, which culminated in the results stated in the table. The net result of these improvements over the last five years is an approach to planning that enables us to project and track each deliverable.

With minor exceptions, each of the 35-plus commitments we made to the customer for product development in 1990 were met within the interval that the customer requested. In addition, each development was completed within the budget our customer was willing to allocate to it.

Hold Gains, Continue Improvement. In 1991, internal customers who were outside the project team's base of process experience placed significant new requirements on the project. Specifically, we were asked to decrease the time required to integrate other manufacturers' products into a new service.

At first, we were unable to maintain a high level of accuracy in our initial predictions of costs, development intervals, etc. for these new requirements. We found that variability in the products from other manufacturers made our planning algorithms inaccurate. By using feedback from the customer and root-cause analysis, we were able to adjust our planning process quickly, so that the plans for the remaining 1991 work items returned to the previous standard for accuracy. (*Root-cause analysis* is a method for determining the underlying reason that a product, process, or service does not conform to requirements.)

Changes of this sort are a way of life in our new business environment. Our challenge is to develop a planning process that allows us to plan accurately for process changes that are outside our experience base.

Conclusion

The AT&T quality approach, which uses PQMI, provided an effective methodology for creating a quality program. It allowed the packet-switching project team to improve the products significantly and reduce the cost structure and cycle time. However, this approach must be driven by a commitment to delighting the customers and must be led by the project's mangement.

As we attempt to replicate these results year after year, we find that continuous quality improvement is difficult to maintain. Although the approach and practices may stay the same, they lose their effectiveness without the focus on customer delight and management leadership. Therefore, other projects that try to emulate these results should pay particular attention to ensuring that the approach, practices, and focus are in place.

Acknowledgments

Several members, past and present, of the packet-switching project contributed significantly to the evolution of the approach described here. Among them are: Bill Bankhead, Marc Braun, Joe Manganello, Joe Paule, Al Rosenberg, Mike Shen, Carl Shu, Bob Stubblefield, and Jeena Srinivas.

References

1. W. E. Deming, *Out of the Crisis*, Center for Advanced Engineering Study, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1982.
2. R. B. Ackerman, R. J. Coleman, E. Leger, and J. C. MacDorman, *Process Quality Management and Improvement Guidelines: Issue 1*, Select Code 500–028, AT&T Customer Information Center, Indianapolis, Indiana, 1987.

(Manuscript received October 17, 1991)

David C. Elmendorf *is a supervisor in the Data Switching, Network Planning and Development Department with AT&T Bell Laboratories in Naperville, Illinois. He is quality manager for the packet-switching development project and is responsible for planning for packet-switching development and network integration. Mr. Elmendorf joined the company in 1980 and has a B.A. in physics from the University of Colorado (Boulder) and an M.S.E. in electrical engineering from Purdue University (West Lafayette, Indiana).*