SOFTWARE PROJECT MANAGEMENT: MOVING BEYOND PROJECT PLANS

Gerald L. Rexing

Gerald L. Rexing *is a supervisor in the International Network Control Point (NCP) and Data Collection System (DCS) Project Department at AT&T Bell Laboratories in Columbus, Ohio. Currently, he is developing software for an international intelligent-network application in Italy. Mr. Rexing joined AT&T in 1976 after earning a B.S.E.E. at Purdue University, West Lafayette, Indiana, and an M.S. in computer science at the University of California, Los Angeles.*

Project managers (PMs) normally regard the creation of project plans as a means to improve software delivery performance. After managing many large software development projects, however, we have found that a project plan should be written *after* the project-management process has been defined. To plan and manage a software project successfully, we must view project management as a process, and the project plan as an activity that prepares data for the process. In our experience, a project plan written early in this process was out of date before it was released, and the project was not managed according to the plan. Recently, our training and experience has focused on a work-breakdown structure (WBS) for managing a project. We have developed methods and responsibility matrices to clarify the tasks associated with project management and to write clear, revisable project plans to fulfill the specific needs of the project-management staff. Thus, we can complete project planning more quickly, define and agree on management techniques, and understand the roles and responsibilities of each member of the project. As a result, we can satisfy our customers by delivering high-quality products on time.

Introduction

The Network Control Point (NCP) is a real-time database system that provides calling services in intelligent networks. An intelligent network is an extension of the basic telephone-service network. Databases in the network contain complex call-handling instructions, which provide services to the users. Since the late 1970s, AT&T has been developing features for the NCP using a highly reliable hardware and

operating system platform. Hundreds of thousands of lines of code have been written to incorporate a broad range of services into intelligent networks sold in the United States and several European countries. Early in the life cycle of the NCP, we developed a few applications for a single customer. We defined project lines for major customer markets and managed them separately as projects. Currently, we are developing several major project lines, each offering numerous services.

More than ever, customers expect high-quality products delivered on time. This emphasis on quality and timeliness, as well as the growth of services, has led us to improve the NCP in many areas, including project planning and management. We have learned a great deal about project-management methods in the four phases of NCP's growth:

- Phase 1—We discovered that project planning was a systematic way of organizing a project's work and describing all aspects of the project's life cycle. At this stage, we learned that a single plan encompassing work activities and contributions from all function groups surpassed the planning efforts of individual groups.
- Phase 2—We focused on customer expectations. Using quality methods and principles, we integrated quality planning and project planning to ensure that customers were satisfied with our products.
- Phase 3—As a result of training we received in project-management methods, we began to focus on project-management techniques as a primary force in developing a project plan. Using those techniques, we collected the specific data we needed to manage the project.
- Phase 4—Our current phase is characterized by project growth and increased leadership in project and quality management, especially by higher management (i.e., department heads and above). We have realigned project and organizational responsibilities to emphasize our projects and customers.

We will continue to use project- and quality-management methods on the NCP as we strive to meet and anticipate our customers' needs.

Our past experiences reveal lessons that others can use to improve their performance. The remainder of this paper examines the phases of the NCP's growth and the lessons we have learned. We attribute the success of our project planning and management to two key factors:

- Project management must be treated as a process, and this process must be identified prior to or early in project planning.
- The data needed in the project-management process must shape the content of the project plan.

Phase 1—Discovering Project Planning

In 1982, project planning gained popularity as a means of improving the delivery of software systems. That same year, the Software Project Management Workshop (SPMW) was formed, and the NCP's project-planning staff began attending the workshop. This intensive five-day project-planning course improved early efforts to develop the NCP by giving the staff:

- A view of what comprehensive planning entailed
- A template for recording project-planning information
- Enthusiasm for the value of project planning.

During the mid-1980s, we put this knowledge to work on the NCP project. Project planning became a team activity; members of several interdisciplinary groups combined to develop a plan that encompassed all activities needed for success. Our project supervisors were convinced that planning the work was important, as much for the value of talking together about the job as for the plan that was produced. The single most important lesson we learned was that all contributors to the planning process must cooperate. However, as we executed the methods recommended, several problems developed that minimized improvement on our project.

First, after completing the SPMW and returning to the work environment, we found it difficult to produce the planning information quickly enough. Project planning was done in addition to our daily work, which made the planning interval longer than the one we had experienced in

the workshop environment. As a result, the plan was in a continuous state of flux, which hampered progress.

Second, the project-plan document was big. We were excited about the prospect of improving our performance and reducing future conflicts by recording all we knew about the project. The SPMW had encouraged us to record our findings, but it was impossible to get the project supervisors to agree on the final content of the document.

Last, we could not agree on a method for breaking the project down into its composite parts and tracking project activities and milestones. The SPMW reviewed basic project planning tools (i.e., program evaluation and review technique, or PERT, and Gantt charts, etc.), but did not give us enough practical training or working knowledge of the tools. The workshop emphasized the production of a plan rather than the management skills and procedures needed to carry out the plan. As a result, we accepted a plan, but had no clear method for managing it. Because we were unable to use the management methods and tools, we reverted to our old methods.

This phase was characterized by:

- Long planning intervals
- Large planning documents that seldom were completed, issued, or updated
- Project management using modification requests (MRs)
- Much day-to-day coordination.

But the learning had begun, and the spirit of cooperation was evident, both in the planning process and throughout the project's life cycle.

Phase 2—Quality Enlightenment

The second phase of development was triggered by a reemphasis on quality at the national, corporate, and project levels. We were still improving the way we applied quality methods; in this phase, we made significant changes in many of our processes, including project planning and management. We studied quality principles, including the writings of Ishakawa,[1] Deming,[2] and Juran.[3] We developed process diagrams, took customer surveys and baseline measurements, and planned quality improvements.

Most of this effort was directed at the software development and maintenance processes, not at project planning and management. Nonetheless, we reached a significant milestone when we developed the project's quality management system (QMS). We recognized that planning and managing quality and planning and managing the project were so closely intertwined that separating them could lead to failure in either one. Therefore, we renamed the NCP QMS as the NCP Management System for Quality to reflect the inclusion of *both* project-management methods and quality management.

Next, we developed a model that integrated our

Panel 1. Abbreviations and Terms

ACWP	actual cost of work performed (cost of achieved activities)
BAC	budget at completion
BCWP	budgeted cost of work performed (planned cost of performed activities)
BCWS	budgeted cost of work scheduled (baseline plan)
EVA	earned-value analysis
FRE	fault-removal efficiency
MR	modification request
NCP	Network Control Point
PC	personal computer
PDCA	plan, do, check, assess
PERT	program evaluation and review technique
PM	project manager
QMS	quality management system
SPMW	Software Project Management Workshop
TAC	time at completion
WBS	work-breakdown structure

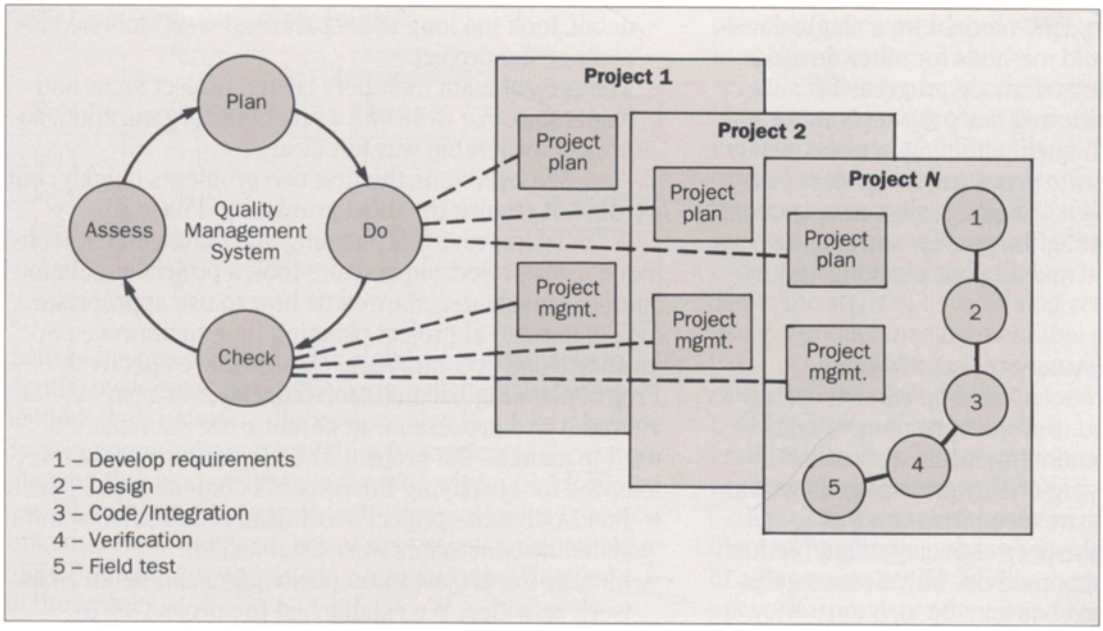

Figure 1. This model illustrates how quality management is connected to project management. Within a single QMS, the PDCA cycle interacts with the project planning and management of numerous projects.

concern for the project and for quality. As Figure 1 illustrates, this model views project planning and management as the connection points between individual project developments and overall quality management. We viewed our QMS as a spiral of "plan, do, check, assess" (PDCA) cycles, in which project planning connects at the D step and project management connects at the C step. Quality *planning* involved organizing our approach to quality (e.g., forming process teams), setting broad objectives (e.g., cost, defect density, etc.), and establishing the baseline development processes. We linked the QMS *do* step with individual project planning. We required that the staff assigned to each project set specific quality objectives, determine process improvements, and establish detailed data collection and reporting plans that aligned with the overall quality objectives of the QMS. For example, the project staff set a specific cost target and defect densities for various phases of the project development. The project staff also planned some process improvements (e.g., code inspection, developer testing, etc.).

Linking project management with the QMS *check* step ensured that actual project data was used to improve the quality of subsequent projects; it also required each project to measure its quality management against the objectives set for it. This model demonstrated the importance of the data used for project planning and management. Since data is essential for quality management and improvement, and quality and project management are closely linked, then data becomes important for project management as well.

The most critical problem we faced in Phase 2 was our inability to turn the quality concepts and models into working methods. We found that learning and application were distinct steps, and the pressure of project schedules only allowed us to apply the principles we learned in the initial phase of the project. Because our department was organized by function and many developers worked concurrently on multiple projects, it

became difficult to change the method for a single development while retaining old methods for other developments. Nevertheless, our staff made progress because we were dedicated to achieving our project-planning and management goals. With quality training in place, we began to define and execute processes for project planning and management. We brought project management under control by integrating the quality- and project-management models and meeting our planning and management goals.

Phase 3—Project Management Drives Planning

In Phase 3, we concentrated on education and project management versus project planning. Quality improvement was our continuing focus; we learned by understanding and applying quality principles, as well as by other methods. While we viewed planning as essential, we discovered that the project-management method should dictate the planning method. The elements that led to improvements were:

- A needs analysis, performed by a project-management consulting firm
- Broad education in project management, presented in an AT&T-sponsored project simulation course
- Project-wide application of project-management tools
- Concentration on the data produced in project planning, not the project-plan document.

We recognized that we were novices at using project-management techniques and that others already had dealt with the problems we faced. The needs analysis conducted for our project identified three major areas for improvement:

- Our project-management methods were not integrated with project planning and did not incorporate the essential elements of basic project management. No WBS was created; the project's PERT diagrams had no specific resource assignments to track and were not used actively in project-management tracking, even for scheduling.
- Our project-plan documents contained some good information but, in general, they contained too much detail, took too long to prepare, and were not used to manage the project.
- The project team members lacked project focus and leadership. We were still a functional organization, so project ownership was not clear.

We overcame the first two problems quickly, but we did not resolve the third problem in Phase 3.

To improve our planning and management techniques, our project supervisors took a project simulation course. This course showed us how to use appropriate data generated at project planning time to improve performance and accommodate change more effectively. Project planning became more concise, because we viewed it as a process for generating the data that we used to manage the project. The planning steps that we followed for specifying the project's content enabled us to:

- Ensure that the project's technical requirements and schedule constraints were established.
- Identify the project's composite parts and detail all its work activities. We established the project's WBS, which is a systematic method of setting priorities for *all* project work required.[4]
- Determine the resources needed for each activity.
- Keeping in mind the need to meet the schedule constraints, schedule the project via task and resource dependencies. [Task dependency exists when two products rely on each other for the activity's product. Resource dependency is based on a single resource being needed (e.g., a person or a laboratory facility) for more than one activity.]

The resulting plan was different from the project-plan documents we had been developing. Its basic components were a project WBS and a project PERT stored in a project-management tool. The project WBS was written as a special memorandum-style document, in which each lowest-level heading corresponded to an activity. In this document, we inserted text at any level that described a set of, or individual, activities. As the key project staff and supervisors wrote this WBS document, they gained a clear understanding of all the elements needed to complete the project. Before the WBS

Table I. Roles/Responsibility Matrix

Role	Responsibility		
	Person 1	Person 2	Person 3
Job 1	—	Primary	
Job 2	Primary	—	Secondary
Job 3	—	Primary	Secondary
⋮			

method became available, the staff had been unable to identify systematically all the work activities that were needed; their planning discussions also had less synergy. By using common memorandum-formatting tools, everyone was able to contribute to the WBS without taking special training. We were able to produce tables of contents that were itemized lists of the project work; we also could produce a complete memorandum that detailed all the tasks.

To create a WBS, the development process must be defined. Most of the activities recorded on a WBS are generated by applying technical requirements to the process used to turn those requirements into a system (i.e., the development process). The more well-defined the development process is, the easier it is to produce a good WBS.

Even so, there can still be disagreements about the "correct" structure of a WBS. The basic disagreement that we had was whether to construct the WBS from the point of view of a system architecture or a development process. We decided to use the development-process viewpoint, because we were applying feature enhancements and additions to an existing architecture. If, however, we had been developing a new architecture, it might have made more sense to use the system-architecture point of view. It would have simplified architecture decomposition into activities to be completed during project development.

As a project team begins to create a WBS, it should identify the top two levels of WBS activities from several alternatives. This approach ensures that project planners have considered all alternatives and agree on how to proceed.

When we began determining resources, we continued to work from the WBS document, simply listing the individuals needed for each activity. Resource conflicts caused network dependencies. For example, Keith is required to complete jobs X and Z; therefore, job X must precede job Z, or vice versa. Relationships involving precedence are recorded in the project-management tool. Thus, the steps of determining resources and scheduling using a project-management tool overlapped.

Our analysis of project-management tools indicated that we should select a tool that would run on our UNIX® System V, to make both the tool and the data generally available to all project-management staff. (UNIX is a registered trademark of UNIX System Laboratories, Inc.)

To organize our project network, we assigned the WBS heading number as the activity number and the heading name as the activity name; we constructed the network from dependency and resource statements that could be extracted automatically from the WBS document. The WBS was our sole planning tool until we were only fine-tuning the project PERT, at which point we made changes directly in the project-management database. From then on, we maintained both the WBS and the project-management database. Along with an action-item management tool, the WBS and the project-management database became the everyday work tools of project management.

This addressed the first two problems identified in the needs analysis. Improvement was evident, both in project performance and in our ability to assess change during the project development. Project management became much more data oriented, and roles and responsibilities were clarified by the use of WBS and responsibility matrices. (A responsibility matrix, illustrated in Table I, identifies the detailed steps to be accomplished in a project, as well as the individuals who have primary or secondary responsibility for each step. It must also include all project-management staff, from department heads to clerical support.)

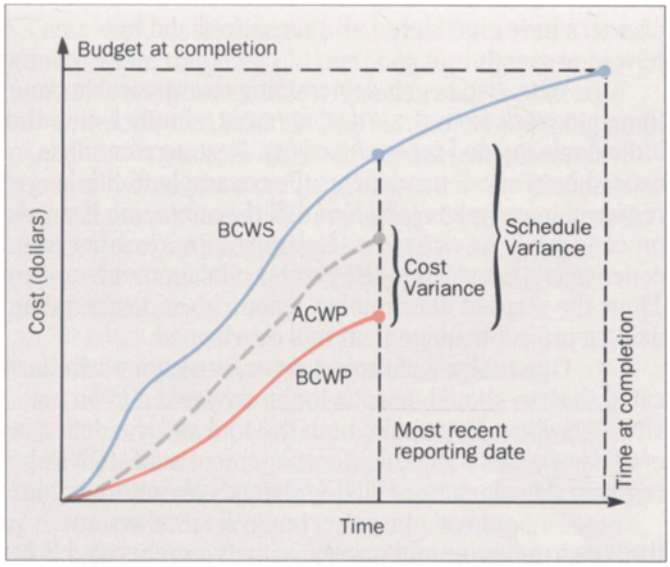

Figure 2. EVA compares a project's cost estimates (which can include staff costs, materials cost, equipment, computer time, and other indirect charges) with the *actual* costs. Cost projections for each activity form the baseline (BCWS). Actual staff effort for each activity yields the actual cost of the work performed (ACWP) and the actual budgeted cost of the work performed (BCWP). The cost variance is the cost difference between the planned and actual costs, while the schedule variance (denoted in dollars, not time) is defined as the difference between the BCWP and the BCWS.

The final problems to be resolved from the needs analysis were project organization and leadership. In Phase 4, we are making strides in these areas and also learning how to plan projects and generate management data.

Phase 4—Leadership Makes It Responsive

Each level of management carries its own scope of control. Up to this point, we had accomplished most of the work and achieved improvement within supervisory and department boundaries. Significant changes were and still are being made by laboratory directors and vice presidents. A major milestone was reached when the directors in our business unit completed their QMS, which outlined specific project-management responsibilities, as well as quality objectives. The business managers thereby demonstrated their understanding of the important link between quality and project management. The result has been more focused project organizations (i.e., "pure" project structures in which all the necessary project resources report to the same management), more systematic quality planning for projects, and more widespread use of classic project-planning and management techniques.

High-level management's preparation of a QMS resulted in more than a written QMS document. Because of their commitment, organizations have been realigned wherever possible to mirror our project developments (which also are aligned to serve our major customers).

We have identified PMs and outlined their specific roles and responsibilities. Now, PMs are required to develop plans and objectives and to report their project and quality progress using standardized reporting mechanisms. Two of these mechanisms, earned-value analysis (EVA),[5,6] and fault-removal efficiency (FRE),[7] have been implemented and have led to improved project-planning and management techniques.

EVA is a method of reporting a project's cost and the progress of its schedule against a baseline plan.[4] This reporting method forces us to define our project in terms of data (see Figure 2 for an example of EVA), including project activities, dependencies, resource allocations, resource costs, and project dates. Identifying the individual resources required for each project activity was the most difficult task, but we have accomplished that task and also have tracked individual resource cost per activity. Now we can produce accurate estimates easily. In addition, periodic reporting and updating have given us a more objective view of project status. Managers can review these project reports

quickly and compare project performance with the planned schedule and costs. Although EVA can view a project's status objectively, it cannot determine which activities are critical. Another management technique, critical-path analysis, fulfills this requirement.

We use critical-path, or slack-time, analysis (the definition of the critical-path analysis technique) to pinpoint activities that are critical to the project schedule. Detailed project planning has uncovered these critical activities, and project tracking using a project database ensures that data remains accurate as the project proceeds. EVA tracks the resource assignments and the cost of activity distribution over time; slack-time analysis requires that we develop a closed network (i.e., a network that produces one end activity), which would improve our dependency planning and analysis. This technique allows PMs to plan and manage the slack time in their schedules. Figure 3 illustrates a method by which the remaining slack time is shown as a function of time to customer delivery. The project manager must ensure that slack time is not used up early in the development cycle.

FRE, a quality metric (see Figure 4), has helped project management to model the development process as a series of fault-insertion and fault-removal steps. The objective of the process is to remove all faults before the product is delivered to the customer. The FRE model estimates the percentage of faults to be removed in each step of the process. Thus, each step has an FRE equal to the ratio of faults removed to faults existing in the product. Project activities are related to a particular fault-removal cycle, which simplifies fault-removal cost accounting.

This metric requires that the development process be controlled and repeatable. It has helped project planning and management by improving the definition of the development process. It also has been instrumental in further clarifying the roles and responsibilities of the development staff, an important element of good project and quality management.

In Phase 4, our business and project managers have instituted all the improvements. Their leadership has made our management methods consistent and responsive.

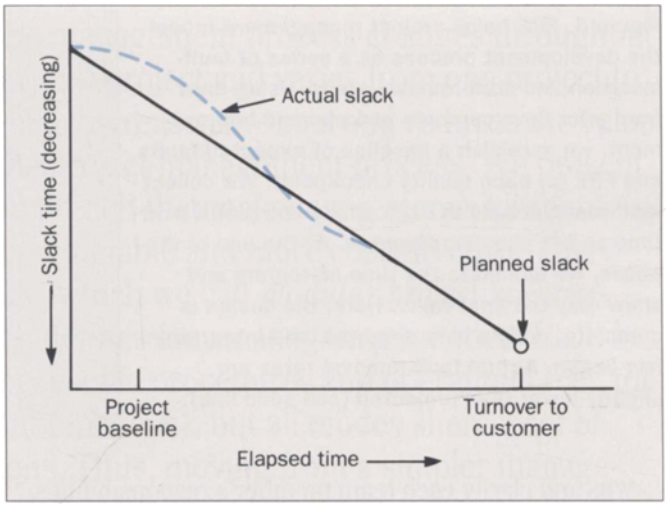

Figure 3. This graph demonstrates how to plan and manage slack time in a schedule. Slack-time management enables PMs to minimize the slack in a schedule and avoid having to delay delivery of the product to the customer.

Conclusion

Over the years, software development and the delivery of many systems have provided numerous opportunities for improvement. Our lessons have been significant and stimulated by our experiences. Specifically, we have a more organized approach to project development, with more predictable results and fewer mistakes; we can predict problem areas; and we have increased customer confidence. As we work on more and larger projects, the following principles of project management will be important to achieving success in the customer's eyes:

- We must understand and define the software-development process at the outset of a project. With those elements in place, it will be easier to create a

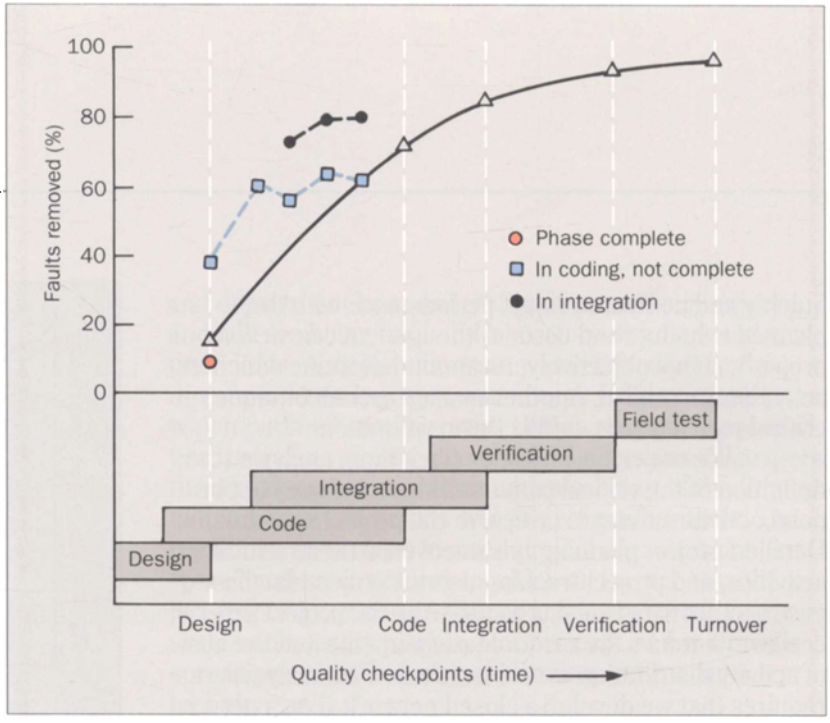

Figure 4. FRE helps project management model the development process as a series of fault-insertion and fault-removal steps. Using data from prior developments and planned improvement, we establish a baseline of expected faults and FRE for each quality checkpoint. We collect fault-removal data in each phase and plot it over time as the phase progresses. At the end of the phase, we eliminate the time histogram and show only the final value. Here, the design is complete, coding is in progress, and integration has begun. Actual fault-removal rates are slightly lower than projected (see solid line).

WBS and clarify each team member's responsibilities.

- Before or while a project is being planned, the management of all people working on the project must agree to incorporate quality management into the project-management process. We must minimize conflicts between groups, and view progress metrics as fair to all.
- Early in the process, we must identify a single individual who is empowered to act as the project manager. Conflicts will always arise, and change will be necessary, but without a single individual who has the power to act, project schedules will slip unnecessarily.
- A project plan must include the data needed to initialize the project-management process. The more important data is to project management, the more objective the process will be. Planning also becomes easier when the required outputs are specified.
- We must institute the discipline to track cost and schedule against the plan and act on the data generated in the process. We should avoid the tendency to view project and quality management as overhead that can be abandoned at the first sign of trouble in the project. Good management processes and data should stand up to this test.

Each software development is unique and will have different requirements, but we have found the principles we described to be critical. As customer expectations change, we must adapt the project and its quality-management processes to meet these new needs. The principles outlined in this paper support and encourage this adaptation for continuous improvement.

References

1. K. Ishakawa, *What Is Total Quality Control? The Japanese Way*, Prentice-Hall, Englewood Cliffs, New Jersey, 1985.
2. W. E. Deming, *Out of the Crisis*, MIT Center for Advanced Engineering Study, Cambridge, Massachusetts, 1982.
3. Juran Institute, Inc., *Seminar on Upper Management And Quality*, Juran Institute, Inc., Wilton, Connecticut, July 1982.
4. D. S. Kezsbom, D. L. Schilling, K. A. Edward, *Dynamic Project Management—A Practical Guide for Managers and Engineers*, John Wiley & Sons, New York, New York, 1989.
5. E. D. Marion and E. J. Riddleberger, "Modular Project Management," *AT&T Technical Journal*, Vol. 70, No. 2, March/April 1991, pp. 49–62.
6. R. C. Vehse, S. F. Nygren, and A. D. Butherus, "Managing an R&D Contract with the Government," *AT&T Technical Journal*, Vol. 70, No. 2, March/April 1991, pp. 84–98.
7. H. Remus,"Integrated Software Validation in the View of Inspections/Reviews," *Proceedings of the Symposium on Software Validation*, Darmstadt, FRG, September 25-30, 1983.

(Manuscript received January 7, 1991)