

THE SOFTWARE DESIGN QUALITY REVIEW

Hosein Fallah and **George T. Tucker** are with AT&T Bell Laboratories in Holmdel, New Jersey. Mr. Fallah is supervisor of the Product Definition Quality Group in the Quality Assurance Center. He is responsible for development and implementation of quality technologies and quality management systems for the front end of the product realization process. In his previous assignment, supervisor of the Design Quality Methods Group, he conducted a number of design quality reviews for AT&T projects. Mr. Fallah joined Bell Laboratories in 1978. He has an M.S. and Ph.D. in applied science from University of Delaware. Mr. Tucker is a member of technical staff in the Systems Quality Department in the Quality Assurance Center. He joined AT&T Bell Laboratories in 1980 and has led a
(continued on page 49)

The AT&T Bell Laboratories Quality Assurance Center conducts Design Quality Reviews of software and hardware development projects, including reviews of specific development products. The objectives of these reviews center around identifying areas of potential quality and productivity improvements. This paper defines the Design Quality Review (DQR), describes the general methodology used in conducting DQRs for software projects, and describes the criteria used to evaluate software projects. Additionally, the paper presents some of the general results and observations from reviews conducted thus far.

Since 1982, the AT&T Bell Laboratories Quality Assurance Center (QAC) has conducted Design Quality Reviews of software and hardware development projects, including reviews of specific development products (e.g., requirements documents). This paper presents the initial procedures developed for these reviews¹ and incorporates the improvements achieved in the procedures since their introduction. It also presents some general results and observations from reviews conducted thus far.

A Design Quality Review (DQR) is an independent evaluation of the development process and associated development products for the purpose of identifying potential quality and productivity improvements. DQRs are different from project management audits. The approach used in these reviews is a departure from the adversary approach generally implied by the term "audit." The distinction between DQRs and project management audits is primarily that of focus: the former focus on the techniques, tools, documents, and procedures through which an idea becomes a supported product, while the latter focus on the organization and management of that whole effort. Further differences are that project management audits are performed early in the project life cycle, but DQRs are conducted throughout the project life cycle; also, DQRs aim at improving quality and productivity in the long term, whereas project management audits are generally conducted

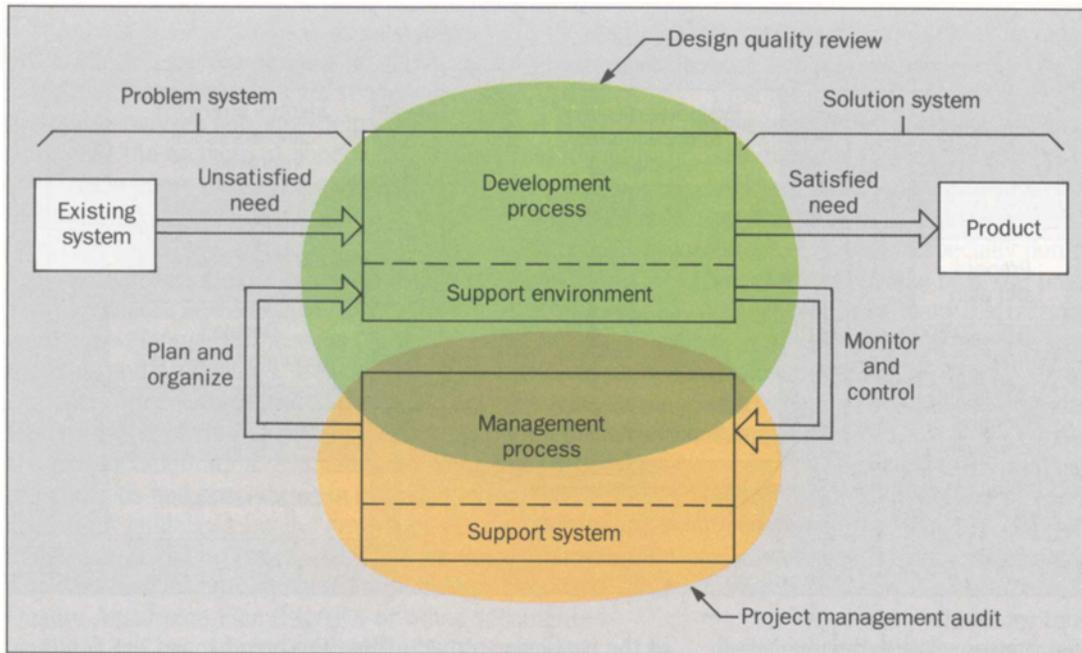


Figure 1. Overview of a development project.

when a project is in trouble and represent a short-term necessity.

However many differences there are between these “reviews” and “audits,” there are also similarities. By their natures, a process and its management are closely coupled so the scope of DQRs and project management audits will overlap somewhat (see Figure 1). Both are performed at the request of project management and both aim at assuring on-time, on-budget, high-quality products. However, the DQRs seem to outshine the project management audits in conducting long-term objective examinations of a project’s processes.

DQR Process

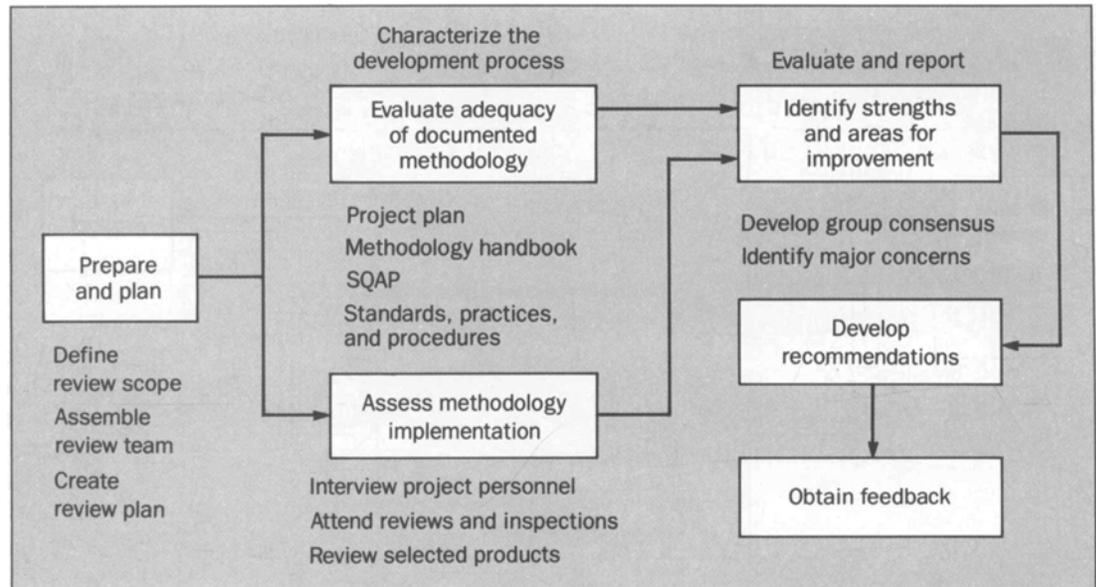
This section describes the generic procedures for conducting DQRs. As shown in Figure 2, a DQR consists of four major steps. These are discussed in the following sections.

Prepare and Plan. A DQR is initiated by a formal request (generally a letter) from the project manager to the Quality Assurance Center. Once a project requests a review, some groundwork is needed before the activities can begin. The pre-review activities are described in the following sections.

Clarify global objectives. Because of the flexibility built into the DQRs, the initial request may not be sufficient to define the task adequately. Thus the first planning activity is to clarify what the customer, usually some level of project management, is seeking through the review. Discussion with the project personnel of the managers’ needs is crucial to the proper initial focus of the effort.

Assemble a review team. A review team will be assembled according to the project size and the particular needs and objectives of the review. Each team will include members with experience and expertise in process reviews, software development, and software quality

Figure 2. The design quality review process.



42

assurance. Additional subject matter experts may be called in as necessary. While the review team is being assembled, the project will identify project contacts to assist the review team.

Understand the project. Understanding the project is a prerequisite to a successful DQR. A tutorial or an overview presentation to the team by the project is a good start. The team also learns about the project by reading project descriptions and other related documents that may be available, and by talking to the project contacts.

Understand the organization. In an effective review, it is crucial to know who is doing what in the project. Organization charts are often outdated and the group titles are generally insufficient for identifying the functions and activities of the project staff. The organization chart is reviewed with the project contacts to identify the functional responsibilities of the people.

Define the review scope. Negotiation of the review's scope with the project's management is the foundation for further activities. Each review is customized to the needs

of the particular project; thus, the breadth and the depth are defined. Depending on the scope, the review may focus on particular phases of the development process. Its depth could vary depending on the review objectives, resource constraints, and timing.

Develop a review plan. The plan is the "road map" for the review. The plan reiterates the purpose of the review and expands on the agreed scope, answering the following questions:

- What will be reviewed?
- Who are the reviewers?
- When can the various review activities be expected to start and finish?
- Where and how will the review activities take place?
- What will be documented in the review report?
- Who should receive the review report?

A draft of the plan is reviewed with the project's management to assure a consensus on the review activities and schedule.

Characterize the Development Process. With the initial groundwork laid, the information gathering activities begin. Within the established scope and time constraints, the team analyzes not only the methodology as it is written down, but the de facto methodology, i.e., the methodology as it exists in practice. The methods of collecting data include analyzing the development process documentation, interviewing project personnel, observing project meetings, and reviewing development process products.

Analyze methodologies. The first step in characterizing the development process is to study the documented methods and procedures. Besides providing a general process overview and background for all the ensuing activities, the review team studies the completeness of life-cycle activity documentation, evaluates the adequacy of the written methods and procedures in providing clear and consistent guidelines for the development process, and formulates an initial hypothesis to follow up during interviews. Any documented development methodology, Software Quality Assurance Plan (SQAP), or other documented practices and procedures are included in this review.

In analyzing development methodologies, the review team looks for task definitions, descriptions, and ownership; solution procedures for tasks; review and inspection procedures; a well-defined life-cycle, broken into activities; well-defined hand-offs between activities; and any standards currently in use. The team wants to look at any related practices and procedures for measurement and tracking, monitoring and control mechanisms, and efficient organizational interfaces. The above lists are not meant to be all-inclusive, but they are a good starting point.

In general, the review team gathers the necessary documents through the project interface contact or the project librarian. The team members review the documents individually and document their observations. These observations are then combined and discussed in a group meeting. The group assesses the adequacy of the documents, and formulates hypotheses regarding the process as it is documented. The detailed documentation

comments are sometimes included in the review report.

Interview developers. The core review is the main time for interviewing developers on site and attending any relevant project meetings. The reviewers interview in subgroups of two in order to provide multiple viewpoints and help balance any individual bias. Using more than two interviewers in an interview may become threatening to the interviewees, especially nonmanagement people. Interviewing people near the project site, but not in their offices, removes many distractions and helps people open up more since they're slightly removed from their usual environment. At the conclusion of each interview, the interviewers document their observations in an interview summary form.

The review team interviews people in the major functional areas of software development included within the scope of the review. A non-inclusive list includes technical staff and their managers responsible for the activities listed in Table I.

A primary reason for interviewing people is to characterize the de facto methodology. Does it follow the documented methodology? Is it consistently applied across people? Is it effective? Additionally, the team wants to gather information to support or refute pending hypotheses, formulate new hypotheses as new information is gathered, characterize the visibility of project tracking and the effectiveness of the project's quality system, and gather upward feedback for the project management. Basic investigative and journalistic interview techniques are the reviewers' tools for this task; the interviewers may use question checklists to prepare for and guide the interview.

Attend meetings. The review team generally has a blanket invitation to any technical, administrative, or informative meeting, either formal or informal, the project holds during the time-frame of the review. The purpose behind attending these gatherings is to observe project communication, characterize general effectiveness of meetings, reviews, and inspections, and characterize compliance of reviews and inspections to any relevant standards.

Table I. Responsibilities of Interviewees

System definition	System requirements
System architecture	Software requirements
Software architecture	Software design
Software code	Software unit test
Software integration test	System test
Customer interface	End-user training
Installation	Software maintenance
Problem reporting and corrective action	Methodology documentation and updates
Project management	Configuration management
Quality control	Developer training

44

The review team, preferably in subgroups, discretely attends and observes various project meetings. Discussing the meeting after it officially ends with some of the attendees is often of value in resolving any questions the reviewers may have. The meetings observed would normally occur during the core review, but they may occur before or after that time also, depending on when representative project meetings are scheduled. The review team is prepared to attend inspections and walk-throughs, as they are scheduled.

Review selected development products. The review team examines a sample of products that represent the various life-cycle activities within the scope of the review. Such examinations serve to check the products' fitness for use in downstream development activities, characterize compliance of the products to existing standards (project or industry), or explore project details for necessary background.

Individual reviewers examine selected products within their field of expertise. These reviews are oriented toward substantive, not editorial or typographical, comments. The team members that examine the same set of documents discuss and integrate their comments.

Evaluate and Report. The team gets together frequently during the core review period and exchanges

observations. These meetings help ensure that the review is progressing according to the plan, provide opportunities to formulate and revise hypotheses, and adjust the planned activities, when necessary. The reviewers may decide to include other people in the list of interviewees, plan to attend additional meetings, or review other relevant documents.

Following the core review, the teams' findings are brought together, combined, evaluated, and documented in a review report. As part of this process, any open items are followed up, and the reviewers obtain additional information and seek clarification on issues that do not have the team's consensus.

In preparing the DQR report the following are considered:

- *Report strengths as well as areas for improvement.* Although the main objective is to identify the areas with potential for quality and productivity improvement, it is equally important to identify those aspects of the process that are done well. This recognizes the successful efforts within the project to produce quality software, aids the project by itemizing things done well that should not be changed for the worse, and helps the team to identify good methods and practices that have potential for use on other projects.
- *Separate major from minor concerns.* The team may find many weak spots. The few major problems are separated from the others and given special emphasis and discussion. This separation assures they do not get lost among other concerns.
- *State the problems succinctly and state why they are problems.* The problems should be communicated to the project very clearly. By stating why the team thinks something is a problem, the team provides the project with the assumptions, perceptions, and the reasoning that led to the team's conclusions. This information helps the project to assess the review findings in light of their more detailed project knowledge.
- *Present recommendations with due caution.* When possible, the team recommends viable solutions to problems

they find while examining the development processes. However, the recommendations are expressed as suggestions that the project may want to consider in developing its own plan of action. The team is aware of the complexities of the technological, functional, and organizational aspects of a development project as they relate to the recommended solutions. While the team's recommendations may provide a good start for addressing a problem area, the people within the project are in a better position to decide what solution is most likely to work.

- *Suggest implementation of key recommendations.* In addition to highlighting the major concerns, the review report summarizes the recommendations to help ensure continual quality and productivity improvement. The review team suggests a priority for the implementation of recommendations, based upon the potential impact of the recommendation, the possible project benefits, and the effort required for implementation.

The DQR report is reviewed with the project management before being finalized. Such a review helps to remove any misunderstandings that the team may have. It also provides an opportunity to find out if there are any major disagreements between the team and the project management and to document the opposing views, if necessary, in the report. With the issuance of the final report, the team formally presents the findings and provides a forum for discussing the recommendations and the project's plan of action.

The report is meant only for the management of the project. It is their prerogative to distribute the report either up or down managerial lines or outside the project. The team members will forward any request for a final report to the appropriate project manager. The team members may report some generalities about the review to their management as needed to keep them informed of the review activities.

Obtain Feedback. It is important to continuously try to improve the review process. The review team conducts a post mortem at the completion of the review. Feedback

from the project also provides valuable information for achieving this objective. A follow-up meeting with the project right after a review may identify some of the changes that could strengthen the review process. The effectiveness of the review, however, can be assessed only over a period of time. Hence, the team leader schedules another follow-up meeting 8 to 12 months after the review or after the product has been released to customers. In this follow-up meeting, the review team will solicit an assessment of the benefits realized (or expected to be realized) by the project so that the cost-effectiveness of the review can be evaluated. The project may also have other suggestions on ways to further improve the review process.

Criteria for Evaluation

As previously described in the process section, a variety of methods and standards are used to evaluate a project. These include corporate and industry standards and development methods, as well as techniques and procedures taught to the developers in courses and workshops. These are described in the following three sections.

Standards. A variety of industry and corporate standards exist which are used as general criteria for evaluating a project during a DQR.

Industry standards. Standards exist within the industry that promote solid engineering practices for the development of software products. The Institute of Electrical and Electronics Engineers (IEEE) has a number of standards in this area. Some that may be used during a DQR include:

- The Standard for Software Quality Assurance Plans (ANSI/IEEE Std 730-1984) provides minimum acceptable requirements for the preparation and content of Software Quality Assurance Plans, and provides a standard against which such plans can be assessed.
- The Standard for Software Configuration Management Plans (ANSI/IEEE Std 828-1983) provides the minimum requirements for preparation and content of Software Configuration Management (SCM) Plans, and is applica-

ble to the entire software life cycle.

- The Standard for Software Test Documentation (ANSI/IEEE Std 829-1983) describes a set of basic test documents associated with software testing, and defines the purpose, outline, and content of each basic document.
- The Guide to Software Requirements Specifications (ANSI/IEEE Std 830-1984) describes approaches to good practice in the specification of software requirements. The use of this guide will help software customers accurately describe what they wish to obtain, and help the software suppliers understand exactly what the customer wants.

Company standards and memoranda. The review team may also use various AT&T Bell Laboratories company standards (area, laboratory, and department level) when conducting a DQR. These could be general internal or technical memoranda. For example, documented guidelines exist for developing a software methodology, writing requirements documents, or developing source code. The use of company or local project standards will depend on the specific needs of a project. In this way, the team may draw on the collective knowledge and experience of the Bell Laboratories community when conducting a DQR.

Software Engineering Methods and Techniques. There are a variety of courses offered within Bell Laboratories that promote software project management methods and software engineering techniques:

- The Project Management Workshop (PMW) helps promote the principles of effective project management. Most Bell Laboratories managers in software development areas have been through this workshop. Some of the many topics discussed in the workshop include the definition of the project life cycle, the identification of project methodology and standards, the establishment of detailed demonstrable project milestones, the identification of the project's critical path, and the definition of the project's organization, with the roles and responsibilities of each organization clearly documented. The PMW handbook provides detailed guidelines and outlines for various functions throughout a project life

cycle. Project plans are reviewed in light of the PMW handbook.

- The Software Engineering Technology Transfer (SETT) Program facilitates the rapid and timely incorporation of state-of-the-art software engineering techniques by Bell Laboratories organizations developing software products and components. It addresses requirements definition, design, testing and inspection techniques, as well as software metrics. The DQR team usually will examine the development process used by the project and will review the project's methods with respect to software engineering techniques, such as those offered by the SETT program.
- The Software Inspection Workshop, which is part of the SETT program, trains participants in software inspections. It teaches the procedures needed to prepare for, participate in, and conduct an inspection meeting. The workshop also includes a discussion of the implementation of inspections in the student's organization. The use of inspections has been recognized as a useful software engineering technique to improve product quality and detect potential problems early in the product realization process. During a DQR, the review team will determine if the project is using the inspection process and will attend inspections that are being held. In light of the guidelines provided by this workshop, the review team will determine if the inspection process used by the project is effective, and may suggest methods for improving it.

Experience. A major strength of the DQR process is the technology transfer that takes place between the QAC and the project being reviewed. This is due to the broad experience and background that exists in the DQR team. The members of the DQR team may be from the QAC, interns on loan to the QAC from other development organizations, or subject matter experts invited to participate in the review. The DQR team brings "real life" perspectives to the review process, experience with existing development tools and methodologies, and the ability to relate and identify with project members during a DQR. This helps to

ensure that recommendations made at the conclusion of a DQR are realistic, and that the project members will “buy into” them, since they often represent ideas from their peers.

General Observations

Strengths. Design Quality Reviews that have been conducted in the past few years reveal several strengths among Bell Laboratories projects. These include a commitment to quality and productivity, a focus on satisfying the customer, and the use of development methodologies, standards, and software project management techniques.

Commitment to quality and productivity. DQRs have revealed that Bell Laboratories projects are genuinely committed to quality and productivity improvement. There is an increasing awareness among project management that quality and productivity go hand in hand, and that an investment in quality improvement will also yield productivity gains.

The evidence that more emphasis is being placed on quality and productivity issues can be found in many ways. Projects are instituting quality improvement programs, through a variety of mechanisms. Several projects have begun to use engineering or quality circles to improve the quality of the development process, the work environment and/or the quality of the product being developed. Other projects have started using inspections and reviews, which are becoming standard practices within the Bell Laboratories community. The use of independent system test groups to improve the quality of the product before it is released to the customer is also practiced by many projects.

Formally documented Software Quality Assurance Plans are being used with more frequency, particularly in larger projects. With the current trend, this may soon become a standard business practice even for small development projects.

Focus on satisfying the customer. The projects that have been reviewed seem intent on customer satisfaction. Systems engineers generally have a close working relation-

ship with the customer. They develop the product requirements either jointly with the customer, or after working closely with the customer and becoming aware of the specific product requirements and customer needs.

Use of development methodologies and standards.

DQRs have shown that in most cases, software development projects within Bell Laboratories follow a documented development methodology. This helps to ensure that a high-quality product is developed, on schedule and within budget. It also provides more uniformity among the product components, and gives project members the guidelines needed to do their jobs.

In addition to using development methodologies, projects are also using other standards and guidelines for specific development activities. For example, most projects use the inspection process for error detection and quality control.

Other standards are being used by projects, to a lesser but ever increasing degree. For example, Configuration Management or Software Test Plans consistent with the IEEE Standards are being developed. Other standards developed within projects, such as software coding or design standards, are also available.

Use of a project management methodology. The Project Management Workshop has been accepted as a de facto Bell Laboratories standard. The projects that have been reviewed have implemented PMW procedures to various degrees. In particular, projects are developing detailed project plans which provide the framework for their development activities (see also the section “Criteria for Evaluation”).

Areas for Improvements. The major thrust of the DQR is to assist software projects in their continual efforts to improve their quality and productivity. Some of the areas with potential for further quality and productivity gain, which are commonly emphasized in these reviews, are described in the following sections.

Quantitative focus on quality improvement. There are wide variations among AT&T projects in terms of quantification of quality measures, goal setting, and tracking

progress. Some projects, especially larger ones, use automated tools, collect and analyze a large quantity of data, and use a variety of in-process metrics; others are less quantitative in their approach to quality improvement. These reviews have served as an effective mechanism for moving software projects toward a more quantitative focus on quality improvement and transferring software quality technology among the projects.

Standardized methodology. In the past, Bell Laboratories projects have developed and implemented their own methodologies. However, significant productivity gains can be realized by standardizing aspects of the development process that are common among projects, perhaps by product type (e.g., switching systems, operations support systems, computers). Some areas of Bell Laboratories are moving toward such standardization. The DQRs provide feedback to the methodologist to ensure that area-wide standards and procedures are widely applicable and have the needed flexibility for adaptation to individual projects.

Tools. An area closely related to methodology standardization, and which needs more attention, is tool procurement and development. A theme often emphasized in these reviews is implementation of more development support tools up-front in the process. Projects are advised against the temptation and challenge of developing their own tools, especially when such tools exist internally in the corporate tool libraries or externally from commercial sources.

Training. A focused approach to training is also an area frequently emphasized in these reviews. The DQRs are helpful in identifying the training needs of the projects to shorten learning curves for new members, reduce errors and rework, and improve the quality of the work environment.

Conclusions

Design Quality Reviews have proven to be a valuable tool for assisting AT&T software development organizations in improving quality and productivity. What follows are some final remarks about the DQRs based on

experience and the feedback received from projects already reviewed.

Customizing the DQR scope and objectives to the specific needs of the project has been valuable in focusing the review effort and is a strength of the DQR process. The people in the projects reviewed have said that, while they do not know how to quantify the impact of the DQR results, they feel that the review benefits are worth the cost of performing the review. One of the reasons why the results cannot be quantified is that the project "cannot see the path that would have been taken if the review had not been performed."

Another strength of the review is that it helps to focus project attention on "some of the important issues at the right time," according to the management of a project recently reviewed. The DQR team can help the project to identify potential problems, since they are not involved in the project and can be independently objective. Specifically, getting attention focused on such areas as project milestones and dependencies (and making them visible) has been identified as possibly the most valuable aspect of a review. Management of one project pointed out that the review "allowed them to anticipate things they wouldn't have foreseen otherwise."

Projects have often commented that while a review is progressing, they see the Hawthorne effect. This is usually defined as the change in people's behavior once they know they are being observed or measured. The result is that improvements in such areas as quality, productivity, etc. occur for a short period because of the participants' enthusiasm, rather than because of some specific process change. The DQR process, however, seems to spawn changes and improvements in a project's development process. Project members begin to look at the things they have been doing and to question existing practices and procedures, which causes beneficial change. In past reviews, many of the concerns identified by the review team have already been addressed by the project before the review is completed.

Through detailed DQR planning, the disruptive

impact a review may have on ongoing project activities is minimized. Projects have also given positive feedback on the policy of disclosing the review findings only to the project management requesting the DQR service, and not to the general Bell Laboratories community. This often removes any political issues that may inhibit these types of efforts. This has helped projects to view the QAC as a partner in their work, and has promoted a healthy relationship in which projects often return to the QAC for additional assistance.

Future Work. Several areas exist for further work, based on previous reviews:

- The ability to measure the effectiveness of the reviews quantitatively. Specifically, determine how the review process impacts a project, and whether there are any “before” and “after” statistics that can be collected and evaluated.
- Define how project quality and productivity improvements can be measured.
- Provide more specific recommendations or alternatives for the project to use in areas where problems or potential problems have been identified. To do this effectively, the knowledge within the DQR team, in areas such as tools, methodologies, and test procedures, should be expanded. Review team members need to be aware of state-of-the-art software engineering techniques, as well as existing tools to support them.

Additionally, more general DQR result information should be disseminated to provide basic “rules for projects

to live by.” Greater emphasis should be placed on documenting where the Bell Laboratories projects are perceived to be now, and what the projects should do to improve in the future.

Acknowledgments

This paper represents the work of many people in the QAC and other Bell Laboratories organizations that has resulted in the evolution of the DQR process. In particular, the authors acknowledge the work of S. G. Crawford in defining the generic review procedures.

Reference

1. S. G. Crawford and M. H. Fallah, “Software Development Process Audits—A General Procedure,” *Proceedings*, 8th International Conference on Software Engineering, London, August 28-30, 1985.

Biographies (continued)

number of design quality reviews of AT&T projects. Mr. Tucker holds A.A.S. and M.S. degrees in computer science from, respectively, Ocean County College and Stevens Institute of Technology. He also holds a B.A. degree in history/political science from Trenton State College.

(Manuscript received February 24, 1986)

MARCH/APRIL 1987 • VOLUME 66 • ISSUE 2