Focus on Network Reliability: Power and Infrastructure

Jeffrey S. Bramley
Nancy B. Irving
Pramod Warty

To ensure the reliable operation of a telecommunications network, power and building systems must be in place, engineered correctly, and well maintained. The Power Quality Resource System (PQRS) supports power systems and other elements of the network infrastructure vital to network reliability programs of the AT&T Network Services Division (NSD). Using best current methods and tools, PQRS delivers a product of "world-class" standards, as evidenced by its absence of defects. This paper describes how PQRS was developed and applied to make AT&T's power and infrastructure provide the most reliable telecommunications network in the world.

Introduction

AT&T operates the largest telecommunications network in the world. It is also the most reliable network in the world. In recent years, there has been increasing emphasis on the role of network reliability in power systems and other network infrastructure elements. While normally maintaining a low profile among the more exciting, high-visibility technologies, power and building systems must be in place, engineered correctly, and maintained appropriately to ensure reliable operation of the network elements they support.

In mid-1992, AT&T NSD, which operates the "core" network, deployed a new application to support network operations and engineering of power and building systems. PQRS plays a vital role in NSD's network reliability programs by supporting numerous maintenance and provisioning processes.

PQRS was developed using the best current methods and tools, resulting in a product of "world class" standards, defined by its delivered quality (i.e., absence of defects). In addition, team productivity metrics far exceed AT&T and industry averages. Viewed as major contributors to these results are application of structured methods using computer-aided software engineering (CASE), application of object-oriented technologies, robust testing using orthogonal arrays, and software reliability engineering (SRE).

Power and Infrastructure

Power and building systems are common systems that provide power and environmental support for the network elements associated with transmission, switching, and signaling. They comprise AT&T's network infrastructure and represent approximately 10 percent of AT&T's investment in the telecommunications network.

A major function of power systems is to convert alternating current (ac), supplied by an electric utility, to direct current (dc), which is required for the proper operation of most telecommunications systems. Many power systems provide reserve energy storage to ensure service continuity if the normal energy supply is interrupted. Power systems also include equipment for power generation, protection, distribution, control, and grounding. Figure 1 is a high-level drawing of the major elements of a central office power system.

Building systems are mechanical systems that provide distribution of air, water, and fuel throughout the building, as well as environmental support, such as control of the temperature, regulation of the humidity in the surrounding air, and air filtration.[1]

Proper planning and engineering, installation, operation, and maintenance of AT&T's network infrastructure are critical to limiting the risk and frequency of failures that affect service.

Panel 1. Abbreviations, Acronyms, and Terms

ac — alternating current
CASE — computer-aided software engineering
CPU — central processing unit
dc — direct current
EASA — Expert Analysis Solution Assistant
NSD — Network Services Division
OATS — Orthogonal Array Testing System
PQRS — Power Quality Resource System
SRE — software reliability engineering
WMS — Work Management System

The Power Quality Resource System

PQRS maintains an inventory of the major units of ac and dc power systems and infrastructure elements, including building environmentals, such as chillers. It supports a critical AT&T business need to maintain complete, accurate technical information on infrastructure elements that can affect service. PQRS is one element in AT&T's wide-ranging work to improve and ensure the reliability of its network infrastructure.

PQRS was developed to support the following processes:

- Demand maintenance — Formulation of appropriate responses to alarm conditions;
- Routine maintenance — Scheduling of routine maintenance work activities via a link to the Work Management System (WMS);
- Capacity planning — Engineering functions;
- Life-cycle management — Programs for replacing, modernizing, and standardizing power and building systems; and
- Environmental Protection Agency record-keeping and reporting.

Today, PQRS provides inventory capabilities for more than 30 individual equipment types or families of equipment; as development continues, the number is expected to exceed 50. A major inventory effort is under way to collect the data for entities critical to the reliable, safe operation of the network. In addition to its inventory function, PQRS provides extensive reporting capabilities and allows power and buildings' support personnel to calculate hours of battery reserve and fuel reserve.

A typical use of PQRS involves support of the demand maintenance process. AT&T's surveillance and alarm center personnel monitor the health of the network and dispatch the on-site work force to respond to alarms. For example, if a power failure alarm is triggered, the center will determine the appropriate response given a number of factors, including battery reserve hours, the presence of a standby engine and its fuel reserve hours, the availability of a portable engine, and so on. All this information can be extracted from PQRS on demand.

In August 1992, PQRS provided Network Operations personnel with key data used in formulating NSD's response to Hurricane Andrew. PQRS's calculations of battery reserve were used to plan and prioritize the deployment of portable standby engines in response to storm-related commercial power outages.

To support the alarm resolution process, current development is linking PQRS with the Expert Analysis Solution Assistant (EASA), an expert system that analyzes specific network conditions, triggered by alarms. Using data from PQRS, EASA will quickly and accurately diagnose the cause of the alarm condition, determine the service potentially at risk, and initiate the required repair work.

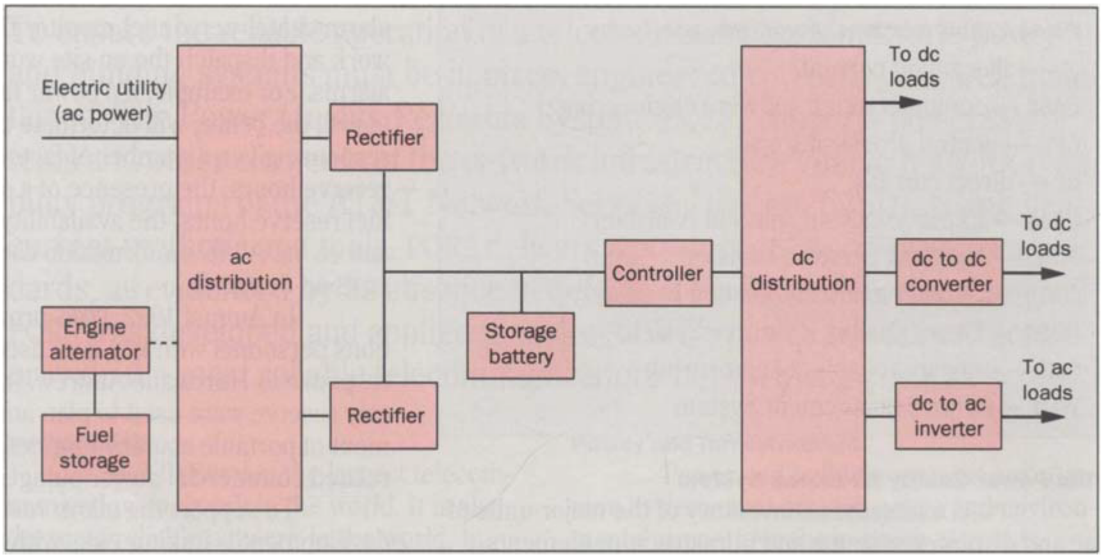
In addition to its support of the demand maintenance process, PQRS schedules routine maintenance for power and building systems. PQRS equipment data is transmitted to the WMS, where required preventive maintenance work activities are derived and scheduled based on the power and building systems inventory for a given location. This automation streamlines the establishment of WMS's power maintenance routines and ensures that key elements of the network infrastructure are being maintained according to corporate specifications.

Engineers also rely on PQRS's inventory for capacity planning and life-cycle management. Engineering includes the evaluation of existing capacities and loads as new loads are introduced to an office, and the design of the optimal solution based on standards, cost, and future growth requirements. PQRS inventory data also supports AT&T's central office record-keeping and reporting of hazardous materials, such as acid in storage batteries and fuel in storage tanks, as required by state and national environmental protection agencies.

The Development Project

PQRS was developed as a "showcase" project within the Operations Technology Center. The PQRS project team created a quality plan early in the development,

Figure 1. The major elements of a central office power system.



applied best current practices and technologies, and collaborated closely with both the user organization and vendors. Because an existing system could support users' needs until the new system was ready, the PQRS project team found significant management support and interest in applying leading-edge methodologies and technologies.

When initially formed, the project team established a set of values to provide an overall direction and philosophy, and to guide day-to-day decisions that would affect the project:

- Planning for quality,
- Partnership,
- Rigorous application of a well-defined development process,
- Effective project management, and
- Simplicity.

Goals for the PQRS project included significant improvements in response time over the "legacy" system, delivery of a "user-friendly" product, and a high degree of reliability/availability given the application's domain of supporting immediate response to alarm conditions and network emergencies.

Figure 2 depicts the standard software development process for the Operations Technology Center. While the PQRS project team applied all the methods implied in the diagram, those that had the greatest impact on the project's productivity and the quality delivered were application of structured analysis and design, using CASE; robust testing, using the Orthogonal Array Testing System (OATS); and SRE.

Structured Analysis and Design

Structured methods provide a systematic approach,[2] a set of procedures and notations for modeling the problem (structured analysis), and the solution (structured design). The project team made an early commitment to use structured methods to engineer the system, and selected Software through Pictures as the analysis and design CASE tool. (Software through Pictures is a registered trademark of Interactive Development Environments, Inc.) The Software Technology Center provided valuable CASE jump-start support.

Because of PQRS's heavy data footprint, the team made extensive use of entity-relationship diagrams to support data analysis and modeling, and dataflow diagrams to illustrate the processes. Following a formal architecture review, design work began with the creation of structure charts.

All the products of structured analysis were inspected, as was each developer's first attempt at a new design or coding challenge. The assortment of technology new to the project team, such as the Daytona database management system (with Cymbal™, its fourth-generation programming language), the JAM user interface builder, and C++, made inspections an important ingredient in the team's quality assurance tactics. (JAM is a registered trademark of JYACC, Inc.)

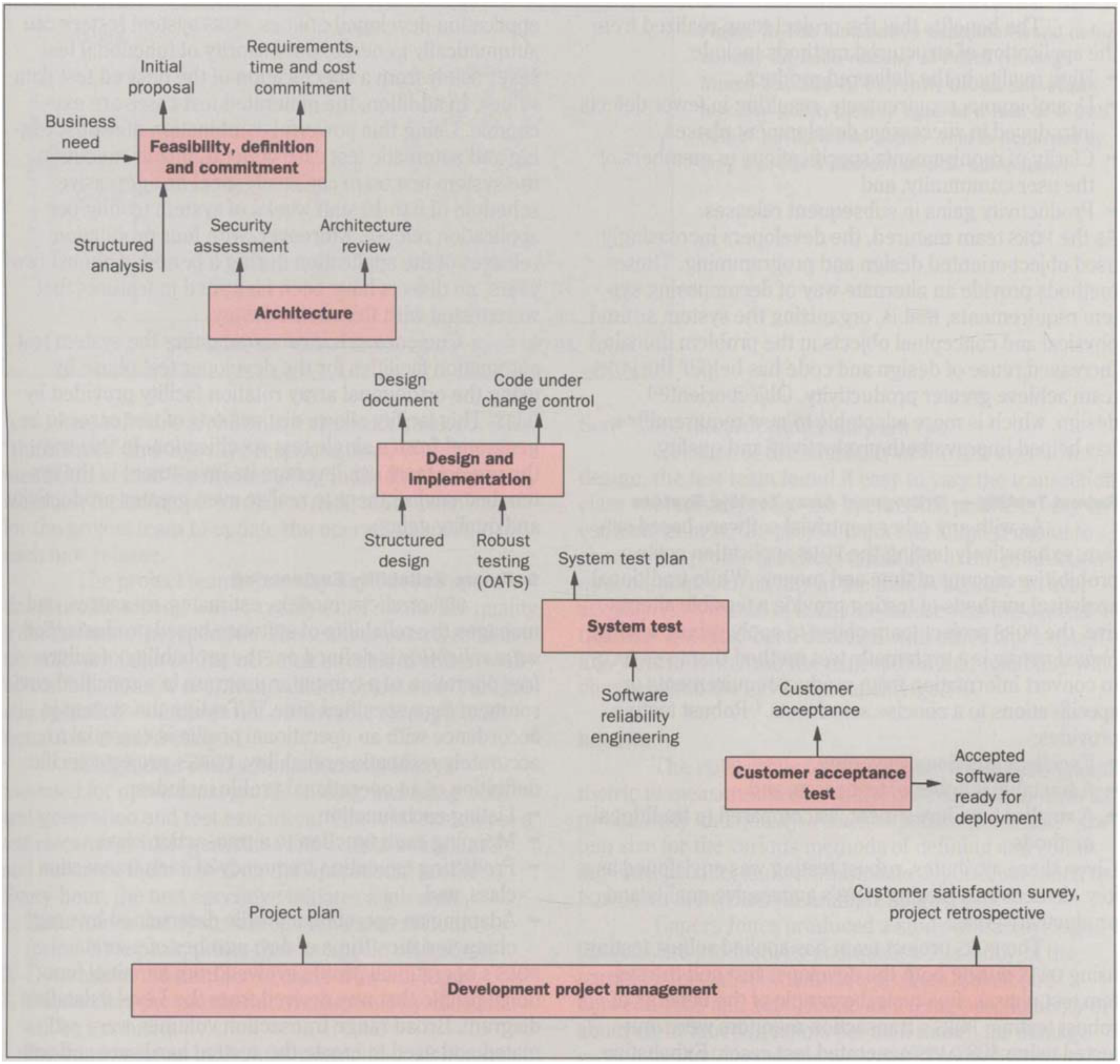

Requirements, time and cost commitment
Initial proposal
Business need
Feasibility, definition and commitment
Structured analysis
Security assessment
Architecture review
Architecture
Design document
Code under change control
Design and Implementation
Structured design
Robust testing (OATS)
System test plan
System test
Software reliability engineering
Customer acceptance
Customer acceptance test
Accepted software ready for deployment
Customer satisfaction survey, project retrospective
Project plan
Development project management

Figure 2. The standard software development process for NSD Operations Systems. The PQRS project team applied all the methods implied in the diagram; however, those with the greatest impact on the project's productivity and quality delivered were the application of structured analysis and design using CASE, robust testing using OATS, and SRE.

The benefits that the project team realized from the application of structured methods include:

- High quality in the delivered product,
- Unambiguous requirements, resulting in fewer defects introduced in successive development phases,
- Clarity of requirements specifications to members of the user community, and
- Productivity gains in subsequent releases.

As the PQRS team matured, the developers increasingly used object-oriented design and programming. These methods provide an alternate way of decomposing system requirements, that is, organizing the system around physical and conceptual objects in the problem domain.[3] Increased reuse of design and code has helped the PQRS team achieve greater productivity. Object-oriented design, which is more adaptable to new requirements, has helped improve both productivity and quality.

Robust Testing — Orthogonal Array Testing System

As with any other nontrivial software-based system, exhaustively testing the PQRS application costs a prohibitive amount of time and money. While traditional analytical methods of testing provide a feasible alternative, the PQRS project team chose to apply robust testing. *Robust testing* is a systematic test method that uses OATS to convert information from product requirements or specifications to a concise set of tests.[4] Robust testing provides:

- Excellent functional coverage,
- A tractable number of test cases, and
- A smaller time investment, as compared to traditional methods.

Given these attributes, robust testing was envisioned as a key element in achieving PQRS's aggressive quality and productivity goals.

The PQRS project team has applied robust testing using OATS during both the developer test and the system test phases. In a typical example of the benefits of robust testing, PQRS's transaction monitors were unit-tested using 408 OATS-generated test cases. Exhaustive testing would have required 1728 test cases. Even with this reduction in the number of test cases, the resultant feature coverage was excellent. No defects were found in the transaction monitors subsequent to the execution of OATS-generated test cases.

Robust testing forms the foundation of the functional test aspect of PQRS system testing. Using OATS and application-developed utilities, PQRS system testers can automatically generate the majority of functional test cases solely from a specification of the desired test data values. In addition, the generated test cases are executable. Using this powerful combination of robust testing and automatic test case generation and execution, the system test team can easily meet its aggressive schedule of 6 to 10 staff weeks of system testing per application release. Moreover, with four production releases of the application during a period of almost two years, no defects have been identified in features that were tested with this methodology.

Current work involves adapting the system test automation facilities for the developer test phase by using the orthogonal array rotation facility provided by OATS. This facility allows distinct sets of test cases to be generated from a single test specification. In this way, the project team can leverage its investment in the system test environment to realize even greater productivity and quality gains.

Software Reliability Engineering

SRE predicts, models, estimates, measures, and manages the reliability of software-based products. *Software reliability* is defined as "the probability of failure-free operation of a computer program in a specified environment for a specified time."[5] Testing the system in accordance with an operational profile is essential to accurately estimating reliability. PQRS's project-specific definition of an operational profile includes:

- Listing each function,
- Mapping each function to a transaction class,
- Predicting execution frequency of each transaction class, and
- Adapting an operational profile determined by usage characteristics (time of day, number of users).

PQRS's operational profile evolved from an initial functional profile that was derived from the Level 0 dataflow diagram. Broad-range transaction volumes were estimated and used to create the system hardware and software architecture. As the software architecture solution was defined, logical groupings of transactions emerged. Because of the difficulty (and inherent inaccuracy) in estimating volumes of individual transactions, the project team agreed to adopt a "transaction class" approach. PQRS's *transaction classes* are groups of transactions roughly equivalent in terms of central processing unit

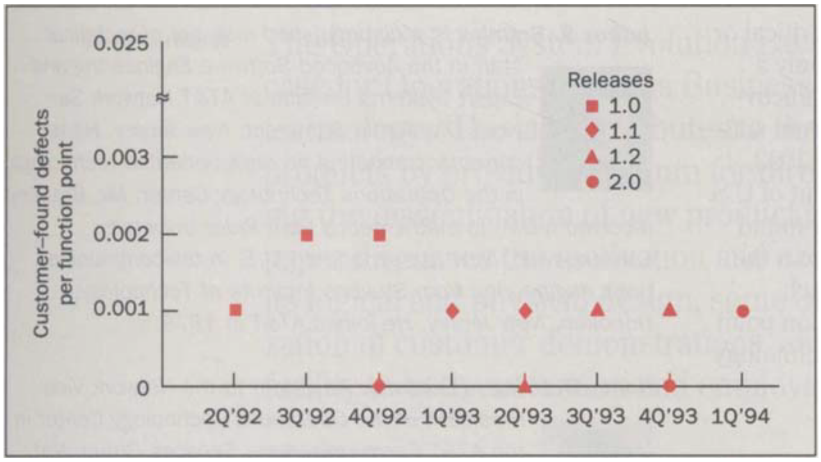


Figure 3. The cumulative customer-found defect density for each release of PQRS (through March 31, 1994), currently 0.001 defect per function point, plotted against a rate of 0.025 defect per function point. This is achieved by only the top 5 percent of U.S. companies.

(CPU) use, software architecture elements involved, user "think time," and objective response time. With instrumentation in PQRS's software to log individual transaction statistics, including performance data, it has been easy for the project team to update the operational profile for each new release.

The project team's primary objective in operational profile testing was to certify PQRS's software quality from the users' perspective. The team expected to identify software failures that occur as a result of either realistic combinations of user transactions or hours of continuous operation — failures not observable through traditional functional testing.

A high degree of automation was always assumed for operational profile testing, including both test generation and test execution. The test team wrote a *test executive*, which generates test scripts using OATS and executes them based on the operational profile. Every hour, the text executive initiates a job and:

1. Determines the system mode for the next hour (prime-time peak, prime-time nonpeak, off hours),
2. Determines the number of users who will log in,
3. Determines the number of transactions by class that will be executed,
4. Randomly selects individual transactions within the class to be executed and generates test cases,
5. Randomly maps test cases to users,
6. Determines a random start time within the hour for each user session, and
7. Creates an execution script for each user.

During each hour, the test executive provides overall control for script execution and monitoring mechanisms. Scripts are discarded after they are run.

Because of the simplicity of the test executive design, the test team found it easy to vary the transaction class volumes. Because the operational profile is only an estimate at best, the project team has found it useful to change the profile to reflect variability in the production environment, i.e., no day in the field is actually an average day. As a result, at least one failure was uncovered that may not have been detected with other types of testing. This follows from the expectation that reliability will change when the environment changes.[5]

Results

The PQRS project team adopted the function point metric to measure the size of the product and, in turn, its productivity and quality. *Function points* "normalize" system size for the various methods of defining and counting lines of code, and accommodate applications, such as PQRS, that are written in multiple languages.[6]

Capers Jones produced a study of U.S. averages for software productivity and quality. His study of the partial history of 4000 software projects, developed between 1950 and 1990, found an average productivity of about eight function points per staff month for management information systems projects.[6] Productivity data for the PQRS project (in function points per staff month) is:

- Release 1.0 — 15.4,
- Release 1.1 — 25.8,
- Release 1.2 — 27.4, and
- Release 2.0 — 31.3.

Experience with PQRS in the field has been impressive. The PQRS project team delivered a "zero-defect" product, one defined as no loss of user functionality; no critical or major defects have been detected at any time. Only a handful of minor, largely cosmetic, failures was uncovered. PQRS has not experienced any downtime from software failures since its initial deployment in May 1992. According to Jones's study, only the top 5 percent of U.S. companies report achieving less than 0.025 user-found defect per function point per year.[6] Figure 3 shows the cumulative customer-found defect density for each release of PQRS, currently 0.001 defect per function point reported, in accordance with the Operations Technology Center metrics program.

Conclusions

Productivity and quality results delivered by the PQRS project team reinforce that "a synergistic combination of defect removal methods will achieve the highest levels of quality and the highest levels of productivity at the same time."[5] The project team also found that application of structured methods and object-oriented technologies contribute significantly to defect prevention.

Power and infrastructure, so critical to the reliability of AT&T's network, are themselves being supported by a system of the highest quality, developed using the best current methods and tools.

References

1. E. J. Kovac, R. J. Skrabal, and J. J. Stockert, "Common Systems," *Engineering and Operations in the Bell System*, AT&T Bell Laboratories, Murray Hill, New Jersey, 1983, pp. 541–567.
2. Roger S. Pressman, *Software Engineering: A Practitioner's Approach*, McGraw-Hill, New York, 1982.
3. Grady Booch, *Object Oriented Design with Applications*, Benjamin/Cummings, Redwood City, California, 1991, p. 15.
4. Robert Brownlie, James Prowse, and Madhav S. Phadke, "Robust Testing of AT&T PMX/Starmail Using OATS," *AT&T Technical Journal*, Vol. 71, No. 3, May/June 1992, pp. 41–47.
5. J. D. Musa, A. Iannino, and K. Okumoto, *Software Reliability: Measurement, Prediction, Application*, McGraw-Hill, New York, 1987.
6. Capers Jones, *Applied Software Measurement*, McGraw-Hill, New York, 1991.

(Manuscript approved May 1994)

Jeffrey S. Bramley *is a distinguished member of technical staff in the Advanced Software Engineering and Expert Systems Division at AT&T Network Services Division in Somerset, New Jersey. He is currently consulting on object-oriented technology in the Operations Technology Center. Mr. Bramley received a B.A. in mathematics from Rider University, Lawrenceville, New Jersey, and an M.E. in telecommunications engineering from Stevens Institute of Technology, Hoboken, New Jersey. He joined AT&T in 1979.*

Nancy B. Irving *is Executive Assistant to the Network Vice President of the Operations Technology Center in the AT&T Communications Services Group, Network Services Division, in Middletown (Red Hill), New Jersey. When this paper was being written, Ms. Irving was the project manager responsible for development of the Power Quality Resource System. She received a B.A. in sociology and psychology from Wellesley College, Massachusetts, and has worked for AT&T for a total of 14 years, from 1972–1981, and from 1989 to the present.*

Pramod Warty *is Chief Technical Officer in the Operations Technology Center at AT&T Communications in Middletown (Red Hill), New Jersey. He is responsible for innovation and deployment of advanced software technologies, including object-oriented technology, software reliability engineering, artificial intelligence, software testing using the orthogonal array method, and computer-aided software engineering (CASE) technology. Mr. Warty received a B.S. in mechanical engineering and an M.S. in systems engineering, both in India. He also earned an M.S. in operations research from Northwestern University, Evanston, Illinois. Mr. Warty joined AT&T in 1978.*