QUALITY TECHNOLOGY IN PRODUCT REALIZATION SYSTEMS

Bruce F. Gundaker, David E. Martinich, and Michael J. Tortorella

AT&T TECHNICAL JOURNAL

Bruce F. Gundaker *is head of the Systems Quality department of AT&T Bell Laboratories in Holmdel, New Jersey.* **David E. Martinich** *is a member of the technical staff in the Quality Theory and Technology department of Bell Laboratories in Holmdel.* **Michael J. Tortorella** *is a distinguished member of the technical staff in the same department in Holmdel. Mr. Gundaker joined AT&T in 1974 and is responsible for quality management methods and tools for the AT&T research and development community. He has a B.S.E.E. from the University of Delaware and an M.S.E.E. from Cornell University. Mr. Martinich joined the company in 1984 and is working on QC Toolkit software, user documentation, and training development. He has a B.A. in mathematics from Carleton College and* (continued on page 20)

This paper focuses on the role that quality technology plays in AT&T's product realization processes and systems. It describes the quality management system within the AT&T research and development community including the eleven principal quality management tasks. There are also descriptions of two software-based quality management tools: the System Used for Prediction and Evaluation of Reliability (SUPER) and the QC Toolkit for statistical quality control.

Introduction

In recent years, AT&T has made major changes in its strategies and structure as well as its role as a world-wide provider of information management and movement products and services. A renewed focus on quality also has accompanied these changes in AT&T's mission and markets. Guided by a corporate quality policy that recognizes quality excellence as "the foundation for the management of our business and the keystone to our overarching goal of customer satisfaction,"[1] people throughout AT&T are working harder than ever to provide products and services that meet the quality expectations of customers.

AT&T's company-wide quality initiative emphasizes the continued improvement of product realization processes and internal business operations. Throughout the corporation, ambitious quality improvement plans and objectives are being established, along with quality management systems to carry out the plans and achieve the objectives. At the same time, AT&T quality professionals have been developing leading-edge quality technology to support the quality management systems and tasks.

The term *quality technology* includes a wide variety of tools, techniques, methods, metrics, guidelines, and procedures that help us to plan, control, and improve the quality of products and services and the efficiency and effectiveness of the product realization process. AT&T is currently engaged in a major effort to expand its quality technology and to put this technology into the hands of marketing, design, development, manufacturing, and business operations organizations.

The Quality Management Functions

Before discussing specific quality methods and tools used by AT&T product realization organizations, we will examine the quality management functions that this technology is intended to support.

Figure 1 shows how a quality management system is being implemented within the AT&T research and development (R&D) community. The system is based on a problem-solving model developed by W. Edwards Deming.[2] The model identifies four elements of a cyclical problem-solving process: *planning, doing, checking, and acting* (shown as the inner ring of Figure 1). In adapting the Deming model to serve as a quality management architecture for design and development activities, we have identified *eleven*, specific quality management tasks (shown as the outer ring of Figure 1) that span the planning-doing-checking-acting cycle.

Planning. Quality management begins with quality planning that includes:

- Establishing specific quality objectives for products, services, and processes
- Preparing comprehensive written plans describing how these objectives will be achieved
- Assigning organizational responsibilities and accountabilities for the identified quality tasks.

Doing. The doing phase of the plan-do-check-act cycle includes identifying and providing needed education and training in quality technology and quality management disciplines; applying documented standards, procedures, methods, and tools to support product realization activities; and implementing a variety of quality controls to ensure that these activities are performed as planned.

Checking. The checking phase of quality management encompasses three major quality tasks: measurement; audits, reviews, and in-process inspections; and quality feedback and reporting. Well-chosen quality metrics can quantify and track the quality of product realization activities and their outputs. From a corporate management perspective, they provide a way to assess collective quality capabilities, a basis from which to set quality improvement goals, and a vehicle for tracking progress towards those goals. At the project level, quality metrics and reports help to identify problems early, pinpoint the root causes of chronic quality problems, monitor the impact of changes made to improve the product realization process, and assess the readiness of products for release to internal or external customers.

Peer-level reviews and inspections of work products provide important checkpoints for finding product defects early in the product realization process. Within AT&T, in-process inspections and reviews are used widely in product definition, software development, and hardware development organizations. In addition, independent audits of product realization and management processes provide timely recommendations for improving development and management methods.

The third element of quality checking is feedback and reporting—providing timely, accurate information about the quality of products and processes to managers and staff at all levels. This feedback can verify that product and process objectives are being met, or alternatively, can indicate where and when improvement and corrective action are needed. It is this feedback about the numbers and types of quality problems that drives the corrective action and improvement process.

Acting. In the acting phase of the cycle, people identify and classify the most important quality problems, diagnose the causes of these problems, and undertake specific improvement projects to eliminate forever the sources of chronic errors and waste.

Supporting Quality Technology

To guide R&D organizations in planning their quality management systems, the Quality Assurance Center at AT&T Bell Laboratories has prepared a comprehensive *Quality Manual*[3] for managers. The manual defines the eleven quality management functions (shown in Figure 1), with suggestions for implementing these functions within various R&D organizations, and advice on the large array of tools and techniques that support implemen-

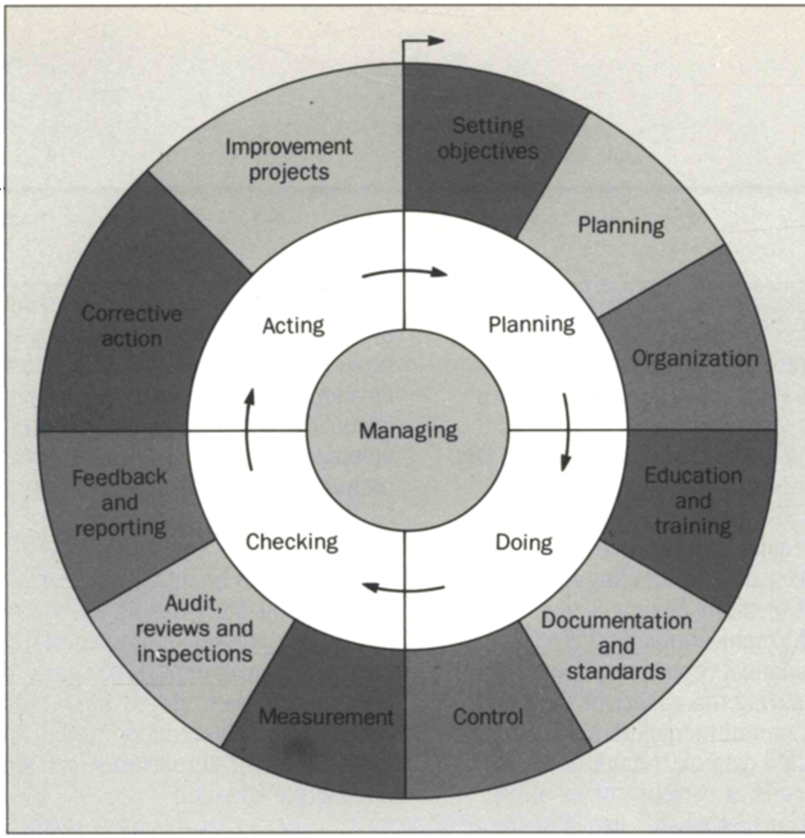


Figure 1. A quality management system for research and development.

tation. The *Quality Manual* and its supplements also provide detailed guidance to R&D organizations on the preparation of project quality plans.

Along with this is a set of guidelines provided by the AT&T Bell Laboratories Business Operations Planning Center. These guidelines span the planning, design, manufacturing, and deployment support phases of the product life cycle. Included is a comprehensive description of the roles, responsibilities, interfaces, handoffs, checkpoints, and activities that make up the process by which high-quality products are designed, manufactured, and deployed.

One powerful technique that is now being used to improve the quality planning and product realization process is *quality function deployment.*[4] Quality function deployment is a structured approach to ensure that customers' needs and expectations are clearly understood and met at all stages of the product definition, design, and manufacturing process. It provides a systematic way of translating customer needs into required quality characteristics and it outlines detailed activities and organizational responsibilities for ensuring that these characteristics are designed into the product and the associated realization process from the outset. Quality function deployment uses seven tools and aids for information gathering and analysis and for group problem solving. With this technique, the effectiveness and efficiency of the product definition process have been improved. Product definitions that better meet the needs and expectations of customers are being produced with less effort and rework.

A wide variety of tools and design aids is available to support the quality checking functions of measurement, audits, reviews, inspections, feedback, and reporting. Since 1983, the AT&T R&D community has used a standard set of broadly applicable *end-product metrics*[5] to track the quality of software. Quality attributes addressed by these standard measures include: the density of faults in completed software products, development organization responsiveness in resolving serious software faults that affect customers, and the impact of software fixes on users. Semiannually, quality results for more than 130 software projects, along with composite summaries and trends, are reported to management. A data analysis and graphics tool called *SQUARES* is used by the Quality Assurance Center to generate semiannual management reports, and is available for projects to use for more frequent internal analysis and reporting.

In 1986, an initial set of *design quality metrics for hardware products*[6] was also standardized and deployed. Patterned after the standard software metrics, the hardware measures track the density of faults in new hardware designs, designer responsiveness in correcting these faults, and the number and cost of resulting design changes. A prototype data analysis and graphics tool (*HQUARES*) has been developed, and a semiannual report established.

In addition to measuring the quality of their end products, design and development managers also need tools for transforming detailed data on the numbers and types of quality problems found at various stages of the development process into information that can help identify root causes and point the road to improvement. In response to that need, the Quality Assurance Center has defined an initial set of *in-process design quality metrics* and an associated analysis and graphics tool called *TIP*. With TIP, designers gain insight into the predominant sources of defects in their processes, understand where in the development process faults are being introduced, assess the effectiveness of in-process review and inspection procedures, and quantify other aspects of development process quality.

A specialized tool developed to aid software developers in quantifying the complexity of software modules is called *CMET*. CMET computes seven complexity metrics for standard C programming language functions, including the numbers of noncommentary sourcelines, executable statements, structure pointers and tokens in the function, and the number of fundamental circuits and unique execution paths in the flow graph. CMET also compares these complexity measures with predetermined threshold values. These comparisons can help designers in identifying error-prone software modules and allocating inspection and testing resources.

Another aid for improving quality management systems is the *independent design quality review*.[7] When project managers request such a review, experienced development and quality professionals examine the adequacy and application of systems engineering, development, and quality management processes and recommend ways to improve quality and productivity.

SUPER

It can be difficult to get accurate information on system availability during the warranty period. In the time period immediately following system installation, system components can experience early life failures at rates often greater than their steady state failure rate. There may be redundant subsystems operating as a safety net that, while increasing overall reliability, can also complicate the job of reliability evaluation.

Standard reliability engineering methods can help solve this problem; however, for complicated systems, these often do not provide enough detail. Even a computer program custom designed with a reliability model for a specific system may not be cost effective if it cannot be reused on other systems.

AT&T Bell Laboratories developed the System Used for Prediction and Evaluation of Reliability (SUPER), a comprehensive software package to create and execute reliability models for general systems using the structural or reliability block diagram modeling technique and the theory of stochastic point processes to describe the evolution in time of these structural models. SUPER handles many complex systems, including those using hot and cold standby redundancy, and many different characterizations of component reliability.

SUPER provides comprehensive reliability models for maintained systems that enable study of both mission reliability as seen by the system user and basic reliability as seen by the system maintainer. SUPER makes it easy to complete the tasks of design for reliability, including the *reliability engineering cycle*, *reliability budgeting*, and *continuous reliability management* during the product design

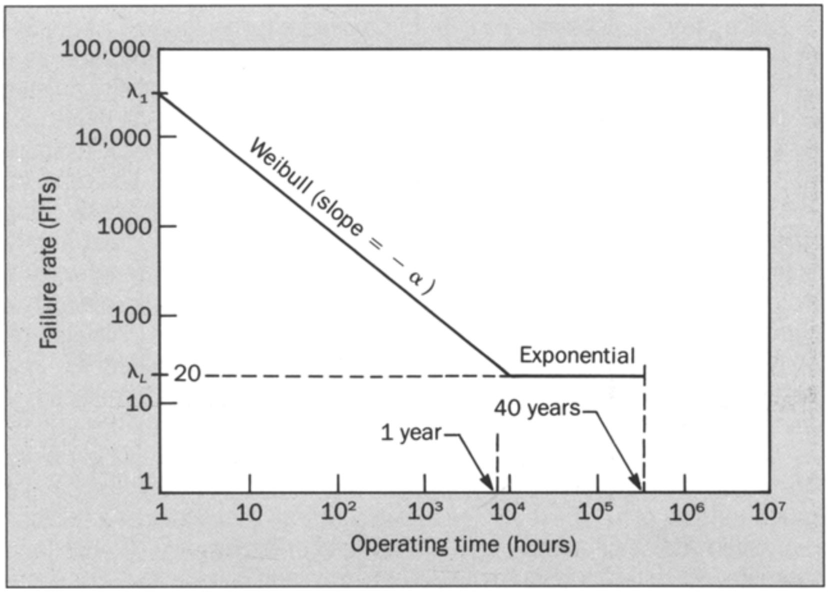


Figure 2. Device failure-rate model. The model is a combination of a Weibull and an exponential distribution. The failure rate unit is in FITs (failures in 10^9 hours).

phase of the product realization process.

Design for Reliability. Design for reliability is a critical need in the product realization process. The most cost-effective reliability improvement steps are those that can be taken early in the design phase of the product realization process. The further a design progresses, the more expensive it is to make design changes that will improve reliability.

The system reliability engineer must begin with system-level reliability objectives that the product is to meet. The engineer needs a way to create high-level system reliability models (even during the exploratory or conceptual phase of system design) to ensure that the completed system will meet these objectives. Design for reliability must start at the earliest phase of system design with models (1) that can be used to qualify the system reliability, (2) that can be augmented to track progress of the design reliability as the design proceeds from concept to hardware and software, and (3) that help create and maintain a system reliability budget.

In addition, circuit pack designers need a way to predict the reliability of the circuit pack, and to check the prediction against the pack's reliability allocation in the system reliability budget. SUPER helps engineers solve these problems by providing reliability modeling capability at all levels of detail.

SUPER and Design for Reliability. Reliability models in the early phases of product design can be created with SUPER's *System Analyst* mode. Reliability models for specific hardware, with detail all the way down to particular components (such as resistors and integrated circuits), can be created with SUPER's *Design Analyst* mode. Intermediate models that the system reliability engineer can use to track reliability progress can be created with SUPER's *Mixed* mode.

To make SUPER most useful for design for reliability, many special features not found in any other reliability prediction software were developed. Some of these are described in this section.

AT&T's component reliability characterization model, based on measured failure rates of electronic components, allows for a larger (but decreasing) component failure rate during the first 10,000 hours of operation than the constant "steady state" failure rate thereafter (referred to as the "device failure-rate model"). (See Figure 2.)

Because of this, the early life reliability characteristics of systems that use these components can be quite different from what is seen in "reliability steady state." Reliability steady state here means that the system reliability process (the stochastic process of successive operation intervals, failure times, and repair intervals) is approximately stationary. More loosely, the effects of early life part reliability and replacement of failed components are smeared out as time passes, and what persists is a much simpler process whose fundamental characteristics are not changed by the further passage of time.

Determining *a priori* when "reliability steady state" begins for systems with any redundancy (electronic switching systems are a good example) is usually difficult. For highly reliable systems (such as satellite or undersea cable systems), "reliability steady state" may not be reached until long after the system's service life is over. So, to see system reliability behavior in early life, it is important to know the reliability characteristics of the system as a function of time. Asymptotic results and limit theorems are not always enough to give the whole picture, especially for studying warranties and other early life issues. Perhaps most important, virtually every AT&T product and system is designed to be maintainable—when a part fails, that part is replaced and the system is returned to service. Computations in reliability models that give reasonable descriptions of a maintained system are generally very hard to find.

SUPER was designed to address each of these problem areas. SUPER automates the AT&T methods for system reliability prediction, and makes it easy to use internal-database component-reliability information for obtaining part reliability parameters by providing automatic lookup by part number and on-line implementation of the procedure for looking up a part by its characteristics.

However, by judicious use of computing power, SUPER takes the reliability engineer beyond merely the automation of manual methods. More complete reliability analyses can now be done, including detailed examination of early life reliability. The time-saving features of the software now make it possible to complete many sensitivity and comparison studies that may not have been undertaken previously because they were too tedious. Further, more realistic models for maintained systems can now be used fruitfully. The point of reliability prediction is to summarize the behavior of the system's reliability process in a concise way that enables timely reliability improvement actions to be taken. SUPER simplifies this task.

SUPER also aids in the job of reliability prediction by making the techniques accessible to persons who don't necessarily have extensive reliability engineering background or experience. It does this by:

- Providing outputs that are simple, intuitive, and easily related to the engineering situation at hand.
- Hiding all its complicated mathematical computations behind a simple-to-use interface that asks for its input information in terms that are easily understandable in the context of the engineering situation.
- Speeding up the reliability prediction process with automatic part list generation from the computer-aided design tools and automatic part reliability parameter lookup in component reliability information databases.

The Reliability Engineering Cycle. The reliability engineering cycle, shown in Figure 3, is so called because its activities repeat until a design qualified for reliability results. By the time one enters the reliability engineering cycle, system-level reliability objectives have been established and accepted by the design organization. The reliability engineering cycle begins with some information on the preliminary design. The preliminary design may be high-level or conceptual at this point; however, the information one normally needs to model the reliability of a beginning design is some system architecture, some description of the reliability of the subsystems, some

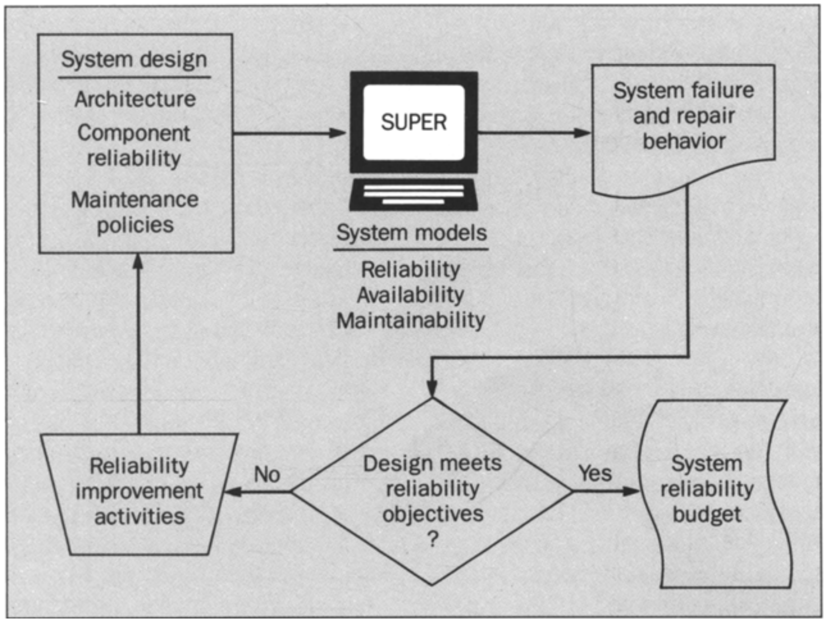


Figure 3. The reliability engineering cycle.

notion of how the system is to be maintained, and some idea of the environmental conditions under which the system will be used. Ideally, this process should be begun even when this information is sketchy. For example, in the design of a disk drive, the design may begin with only the information that the system will consist of a drive unit, a controller, and a power supply. While this is enough information to begin the reliability engineering cycle, as this information becomes amplified and refined, the reliability models can be maintained and updated as new information becomes available, creating better information for reliability improvement decisions.

After the preliminary design information is captured, it is put into SUPER. The reliability, availability, and maintainability models in the program are tailored to the design, and SUPER produces from them a description of the system's future failure and repair behavior. This description is then compared to the system reliability objectives to determine if the current version of the design is qualified (i.e., capable of meeting the overall system reliability objectives). If not, reliability *improvement* activities must be undertaken to increase the reliability of the design. For our purposes here, these fall into four major categories: strengthening the existing design, redesigning part or all of the system, eliminating known causes of failure, and starting reliability growth programs. Any of these reliability improvement activities can modify the original design.

These modifications are then captured in the reliability model previously constructed with SUPER, and the new model then produces its description of the system reliability based on the design modifications. Again, one compares these results with the overall system reliability objectives, continuing the process of reliability improvement, modeling, and comparison until the design is qualified.

During this application of the reliability engineering cycle, SUPER would most often be used in the System

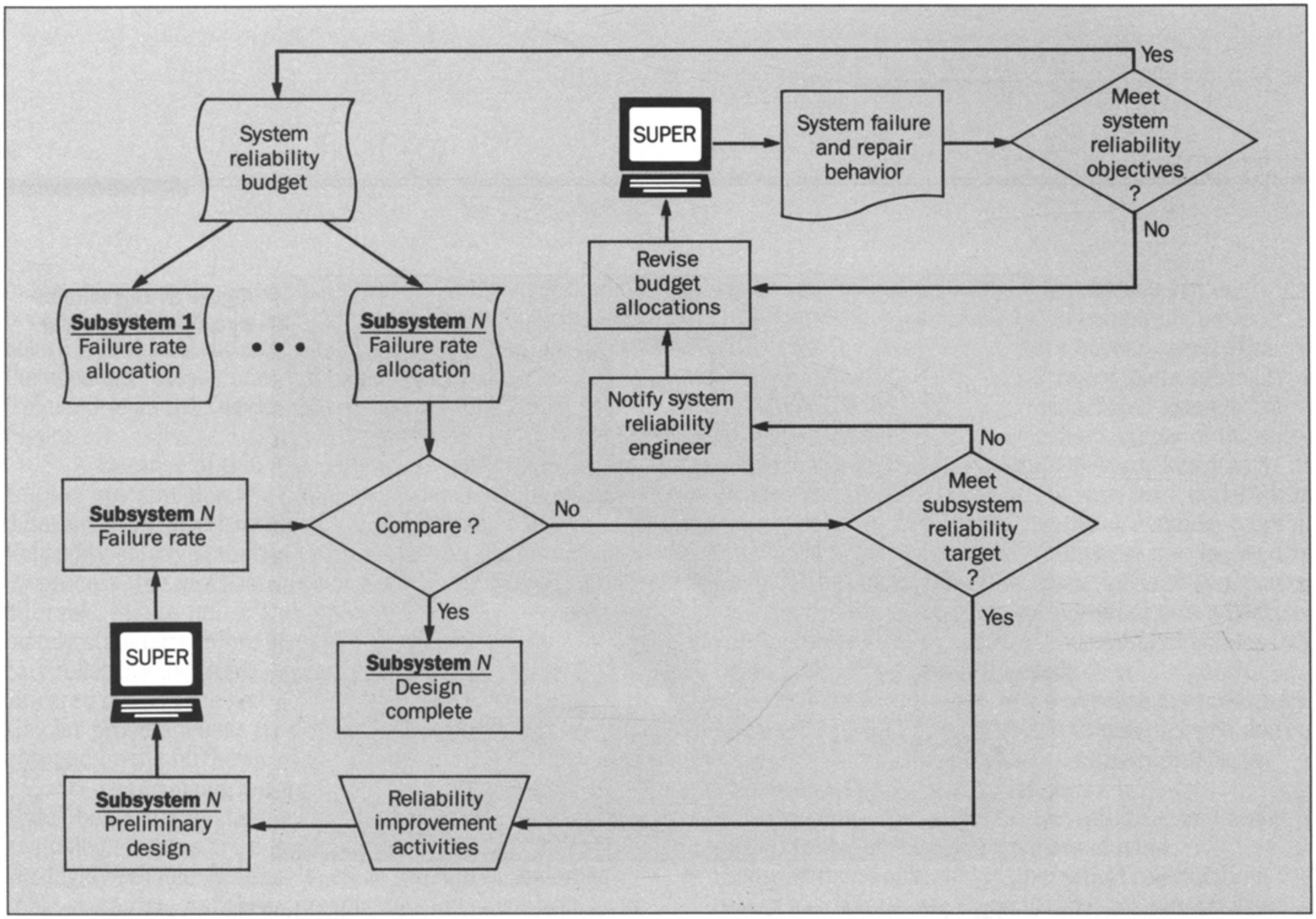


Figure 4. Continuous reliability monitoring.

Analyst (SA) mode. A systems analysis reliability study typically is characterized by high-level designs, little specific hardware or software information, rapidly changing design possibilities (i.e., a general, exploratory stage).

These studies should be done early in the design phase of the product realization process. The system analyst has to study the effects of different reliability architectures (redundancy, distributed architecture, maintenance policy, etc.), and possible subsystem, or block, reliabilities. SUPER is used in the SA mode by creating a reliability block diagram (RBD) at the appropriate level of detail (at this stage, this may be driven by the amount of subsystem reliability information at hand) and by choosing from among the list of life distributions built into SUPER to describe the reliability of the blocks. If life distribution information for a block is available in the form of a table, this can also be used directly. Appropriate reliability models can be executed, and sensitivity studies can be run to see the effect of possible changes in architecture, subsystem reliability, maintenance policy, or environmental conditions.

A major benefit of the SUPER interface is that it enables system changes to be made and sensitivity studies to be run interactively in a single SUPER session. This is a real time saver, and contributes materially to the design of more reliable products.

Reliability Budgeting. After the design has been qualified in the reliability engineering cycle, we can issue the project system reliability budget. (See Figure 3.) The system reliability budget is a breakdown of the overall system reliability objective into reliability allocations for subsystems and components, arranged so that if the budget allocation for each of the subsystems is met, then the overall system reliability objective will be met.

This breakdown is necessary because the *overall* system reliability objective may not be useful to individual subsystem design areas. For example, a requirement on system availability would not be meaningful to the designer of a subsystem or circuit pack in the system. This designer typically does not have access to the same total system information that the system reliability engineer does, and so is unable to judge the reliability of the individual subsystem in relation to that of the entire system.

Budget allocations should be made in terms the subsystem and component designers can understand. For example, component reliability allocations can be made in terms of a maximum failure rate for the subsystem, viewed as a nonmaintained system, over the system service life. This is a meaningful concept that designers understand and it is compatible with the reliability modeling methods used during the reliability engineering cycle.

Reliability budgeting is largely a process of "working the reliability prediction model backwards." In operational terms, this means that tentative reliability allocations for subsystems and components are already present in the reliability prediction models used up to this point. These come from the subsystem reliability information needed for SUPER. Preparing the reliability budget is then mostly a matter of refining previous prediction models to account for any new information about the qualified final design. The reliability budget should not be a revolutionary change at this late stage.

Continuous Reliability Monitoring. Once the system reliability budget is established, designers can use SUPER to verify that their subsystems meet the reliability budget allocations established for them. In addition, the system reliability engineer can incorporate the latest design information into the overall system model as a way of keeping the system reliability predictions current with the design.

For this task, the subsystem designer uses SUPER in what is called the Design Analyst (DA) mode. Here SUPER is used to make reliability predictions about a specific piece of hardware, such as a circuit pack. These studies are characterized by very specific part information, and may include individual part temperatures and electrical stresses. SUPER uses the device failure-rate model to characterize part reliability. It also includes a convenient spreadsheet interface for these studies that shows the effects of varying part reliability parameters, temperature, electrical stress, quality levels, and other factors in real time.

In the DA mode, the circuit pack, for example, is modeled as a single block, with its reliability derived from the list of parts on the pack and their reliability parameters. Nonmaintained system models are most often used here, because the circuit pack is usually a part of a larger system that may be maintainable. From these, one obtains life (time-to-failure) distribution information about the pack, as well as pack failure-rate information, as a function of time. This information can be compared to the system reliability allocation for the subsystem as shown in Figure 4.

The Figure shows that if the result of this comparison is unsatisfactory, two courses of action are available. First, the designer can carry out appropriate reliability improvement activities for the subsystem or circuit pack. (This resembles a reliability engineering cycle in miniature for this subsystem alone.) If it is not possible or economical to achieve the reliability called for in the system reliability budget for this subsystem, the system reliability engineer must be notified and a revision of the system reliability budget attempted. This revision amounts to a redistribution of reliability allocations among subsystems to try to enlarge the allocations for those subsystems that are having difficulty meeting the original ones, and tighten the allocations for those subsystems meeting their original targets easily.

The information generated by the various subsystem designers can be combined in the system reliability model to produce updated reliability predictions for the entire system as the design progresses (using nonmaintained or maintained system models, as appropriate). In this way, the system reliability prediction is continuously monitored so that there are early warnings of possible problems. Designers should be made responsible for communicating updated reliability predictions for their subsystems to the system reliability engineer. Designers can be trained to use SUPER to make predictions more quickly and easily.

During the evolution of the design from concept to complete hardware and software specification, the system reliability engineer uses SUPER in Mixed mode. Here, the SUPER model has some blocks in the reliability block diagram specified by part lists, and some by the life distributions characteristic of SA mode. There is no restriction on the level of detail that can be incorporated into a single SUPER model. Blocks described by parts lists and blocks described by built-in distributions can coexist in the same model. Mixed-mode operation is a good way to keep reliability models current with system development as it progresses from the less defined system analysis phase to the complete detailed system design.

With continuous monitoring and sufficient concern for prevention and solid reliability engineering the first time through, it should be possible to completely avoid last-minute "reliability improvements."

To summarize, SUPER is used in the SA mode to prepare the reliability budget, and in the Mixed mode to keep the budget up to date and support subsystem reliability tradeoff activities. SUPER is used in the DA mode by subsystem designers to check that they are meeting their budget allocations, and can also be used this way on the complete system to verify that overall reliability objectives are met. The DA mode on the whole system can also be used to obtain more detailed information about the early life reliability of the entire system.

SUPER and Computer-Aided Design Systems. To promote more effective use of design for reliability in the product realization process, SUPER is currently being integrated into the AT&T computer-aided design system. A major objective of this integration is to make circuit-pack reliability prediction in general (and use of SUPER in particular) a routine part of engineers' activities during the design of a circuit pack or subsystem. Essentially, integration consists of setting up automatic pathways for the flow of information to and from the computer-aided design system, SUPER, and component reliability databases. (See Figure 5.)

To run a Design Analyst reliability prediction for a circuit pack, or some other subsystem described by a list of parts or components, SUPER needs the parts list as basic input. It is much easier if this parts list, which is available in the computer-aided design system, can be automatically transferred to SUPER. Once integration is complete, the SUPER DA mode will be especially easy to use because circuit pack parts lists will be available to SUPER automatically, with no manual intervention.

Another advantage of integration is automatic lookup of component reliability parameters in a reliability database. (This is also illustrated in Figure 5.) SUPER passes the parts list for the circuit pack or subsystem over to the reliability database, which then matches the part numbers with its own internal information. The database then returns to SUPER reliability parameters (such as the parameters of the device failure-rate model) for each of the components it has found.

Both automatic pathways save time and energy in avoiding manual entry of parts list data and component reliability parameters. This further advances the goal of making reliability prediction, and design for reliability in general, an important contributor to a successful product realization process.

QC Toolkit

Statistical Quality Control (SQC) is the application of statistical principles and techniques for monitoring, controlling, and improving the quality of design, manufacturing, business, and maintenance processes. SQC techniques are, for the most part, simple but tedious to do by hand. Such a

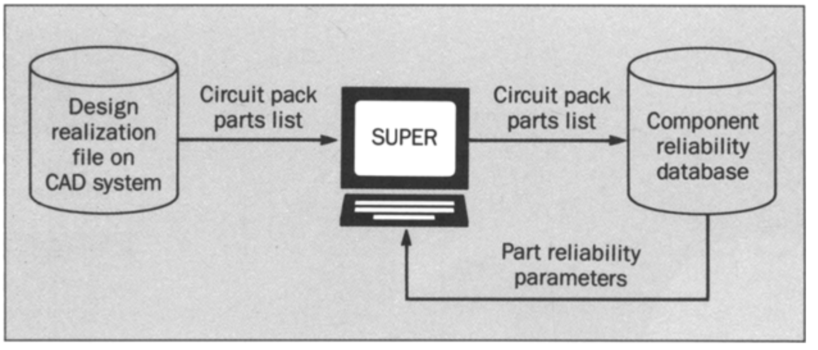


Figure 5. Automatic links in SUPER integrated with a CAD system. CAD = computer-aided design.

situation is ideal for software applications. One advantage of using quality control software is that, with minimal training, SQC techniques can be quickly placed into the hands of a large number of people. One benefit of this is that SQC will be used by more people and applied to new areas; and thus, more people realize that they can do something about quality, more people start thinking about what they can do to improve quality, and more people start trying to improve quality.

As a part of its efforts to support quality control and improvement throughout AT&T, the Bell Laboratories Quality Assurance Center has developed a powerful and flexible software tool for SQC. Called *QC Toolkit*, this tool runs on the AT&T PC 6300 and PC 6300 PLUS, and can be integrated with existing and planned factory information systems.

QC Toolkit is designed to monitor, control, and improve manufacturing and business processes using the tools and techniques found in the *AT&T Statistical Quality Control Handbook*.[8] The *Handbook* was written about thirty years ago by AT&T personnel to provide a guide for applying SQC principles to AT&T manufacturing operations. The *Handbook* is used extensively within AT&T and other companies all over the world, and it is widely viewed as one of the best books on SQC.

Users of QC Toolkit (and the *Handbook*) include: Quality Control (QC) teams (a *quality engineer*, a *product* or *process engineer*, and a *shop supervisor*). The activities of QC teams can be described by the QC cycle. (See Figure 6.)

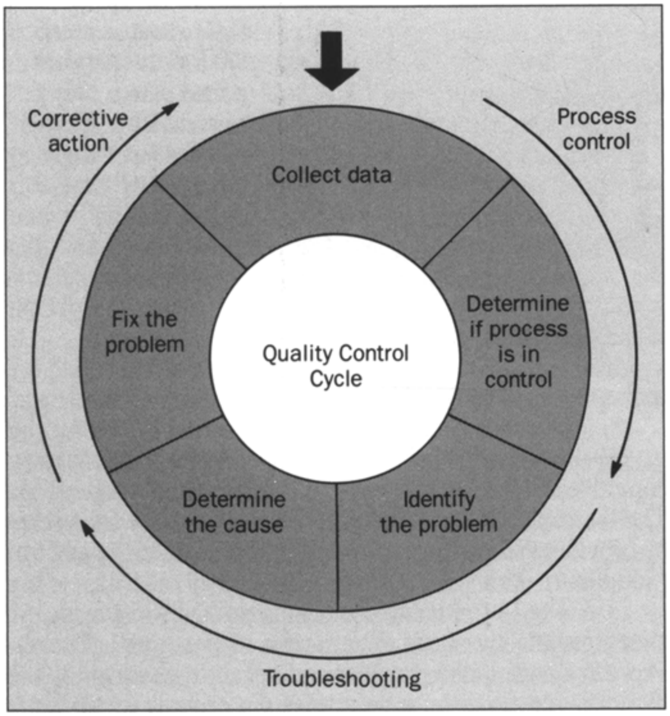
QC Toolkit supports the activities of QC teams by having the QC cycle built into the software. QC Toolkit is comprised of four subsystems called *folios*. (See Figure 7.) The four folios support the first four steps in the QC cycle by providing statistical tables and high-resolution graphs. (See Figure 8.)

The Data Management folio provides a spreadsheet for data entry and tools for file management. The Process Control folio provides control charts with automatic diagnostic tests to determine if a process is out-of-control. The Problem Identification folio provides Pareto diagrams, histograms, and normal probability plots to see what kind of problem exists. The Problem Diagnosis folio provides scatterplots, boxplots, and analysis of means to determine the causes of problems.

QC Toolkit uses a windowed interface that provides guidance and flexibility, so that software can be easily used by both experts and novices in SQC.

QC Toolkit in a Factory Environment. As a PC-based tool, QC Toolkit can be used as either a stand-alone SQC software system (for those locations without factory information systems) or as a part of an integrated quality system.

QC Toolkit as a stand-alone system. As a stand-alone, QC Toolkit can support data entry and data management, as well as all other SQC functions. Each shop (or line or cell) has its own PC. QC teams use QC Toolkit to manage



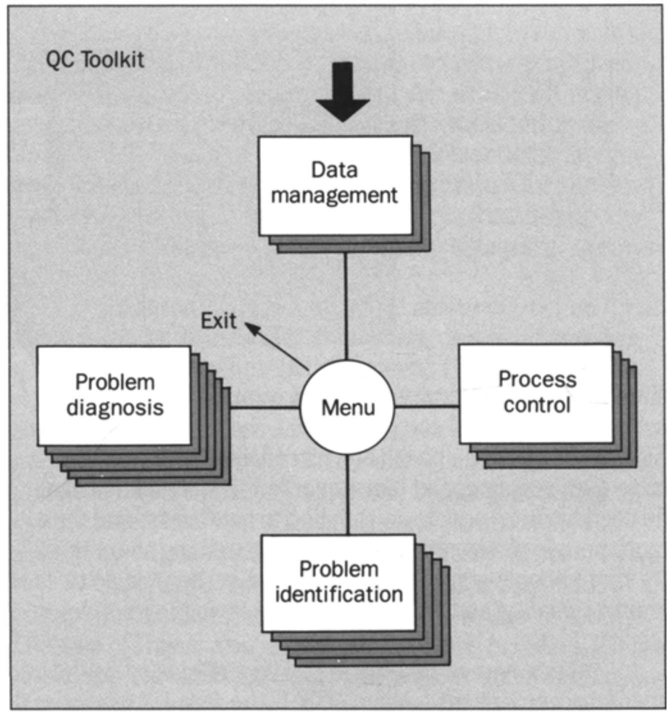
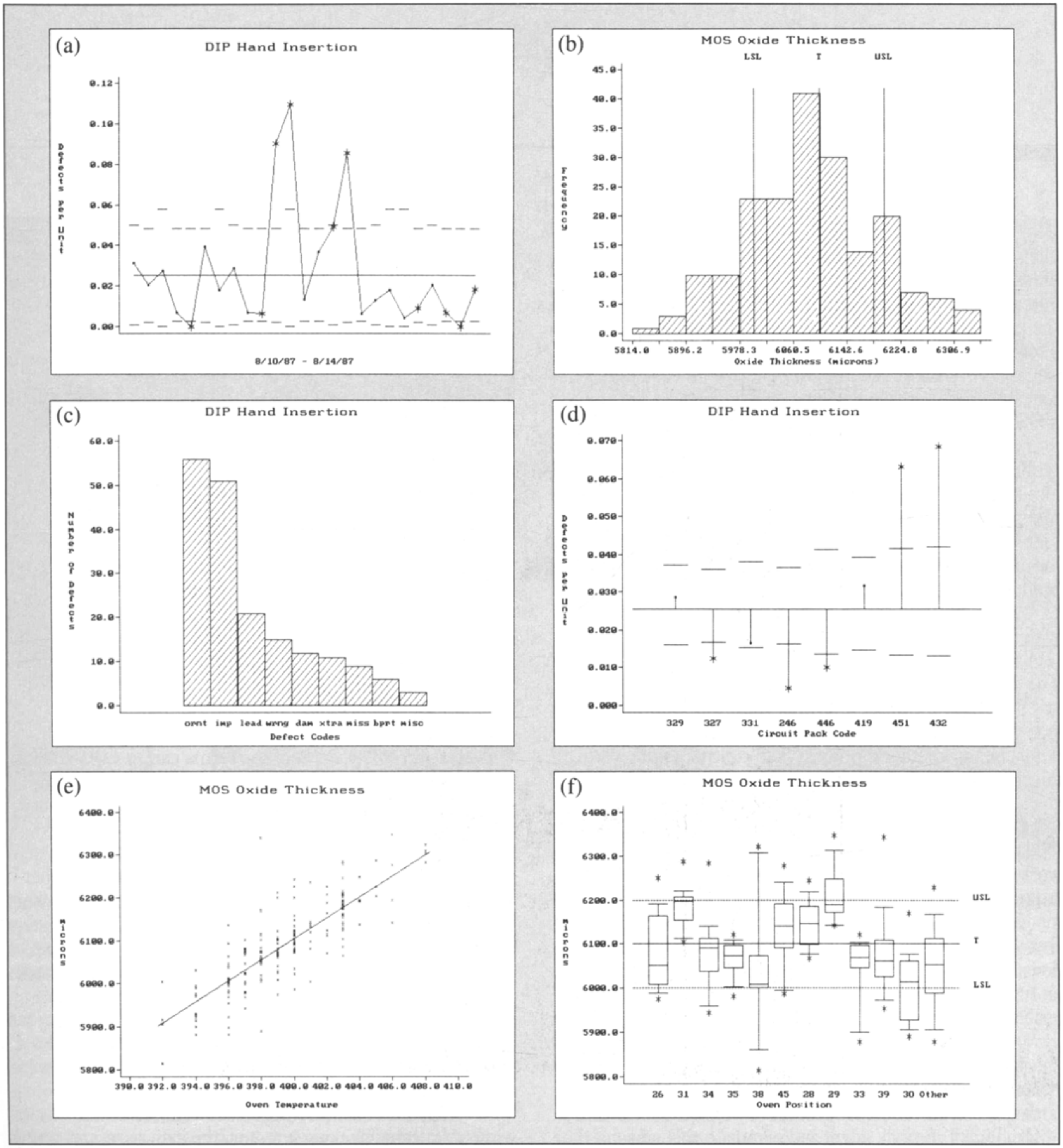


Figure 6. The Quality Control cycle. Data are collected and analyzed to determine whether the process is out of control. If so, the Quality Control team first identifies the problem, then diagnoses it. Once the cause has been determined, corrective action is taken.

Figure 7. QC Toolkit. The four QC Toolkit folios are designed to guide the Quality Control team through the first four steps of the Quality Control cycle. Each QC Toolkit folio contains statistical and graphical tools to support the corresponding step in the cycle.

a database in the form of a collection of ASCII files. (ASCII stands for American Standard Code for Information Interchange.) The data from each process can be entered into a file either through QC Toolkit (manually, through the spreadsheet data editor) or directly into the file (automatically, by bar code reader, for example). (See Figure 9.) These data are analyzed and stored on the PC; data communications can be handled either by a local area network or by floppy disk.

Figure 8. QC Toolkit provides output in easy-to-read well-formatted tables and graphs that can be displayed on the screen or printed. Shown are: (a) control chart, (b) histogram, (c) Pareto diagram, (d)analysis of means, (e) scatterplot, and (f) boxplots.


(a)
DIP Hand Insertion
Defects per Unit
8/10/87 - 8/14/87
(b)
MOS Oxide Thickness
LSL
T
USL
Frequency
5814.0
5896.2
5978.3
6060.5
6142.6
6224.8
6306.9
Oxide Thickness (microns)
(c)
DIP Hand Insertion
Number of Defects
ornt
imp
lead
wrng
dam
xtra
miss
bprt
misc
Defect Codes
(d)
DIP Hand Insertion
Defects per Unit
329
327
331
246
446
419
451
432
Circuit Pack Code
(e)
MOS Oxide Thickness
Microns
Oven Temperature
(f)
MOS Oxide Thickness
Microns
USL
T
LSL
26
31
34
35
38
45
28
29
33
39
30
Other
Oven Position

Factories that have information systems for handling and storing process data use QC Toolkit in a much different way. The information system runs on a host computer (an AT&T 3B computer, for example). This is where the process data are stored as well. In this setting, QC Toolkit is not used for data storage.

As a part of an integrated factory information system, QC Toolkit's role depends on whether the factory already has a process checking system (a software system for entering process data into the overall information system). AT&T manufacturing locations employ a variety of locally developed process checking systems. These systems are used for data entry and process checking (determining whether a process is out-of-control), performing validity checks on data, managing databases, and producing reports.

QC Toolkit and process checking systems. In factories with process checking systems, the process checking system and QC Toolkit are complementary. (See Figure 10.) The data are entered through the process checking system and stored on the host computer. Here, the process checking system is used for data entry and validation, for determining whether a process is in control, and for producing reports.

In addition, each QC team has its own PC running QC Toolkit. This PC is used as an *SQC workstation*. The PC is connected to the host computer (via a Datakit™ network, for example) and can be in either terminal mode logged onto the host, or in PC mode. While in the terminal mode, the QC Toolkit user accesses the quality information database, obtains the desired data for analysis, and downloads this data onto the PC hard disk in the form of an ASCII file. The user then switches to PC mode, and analyzes the data using QC Toolkit.

QC Toolkit complements the existing process checking system by supporting such activities as:

- *Process troubleshooting.* If the process checking system detects an out-of-control condition, QC teams can use QC Toolkit to track down and eliminate the cause of the problem quickly.

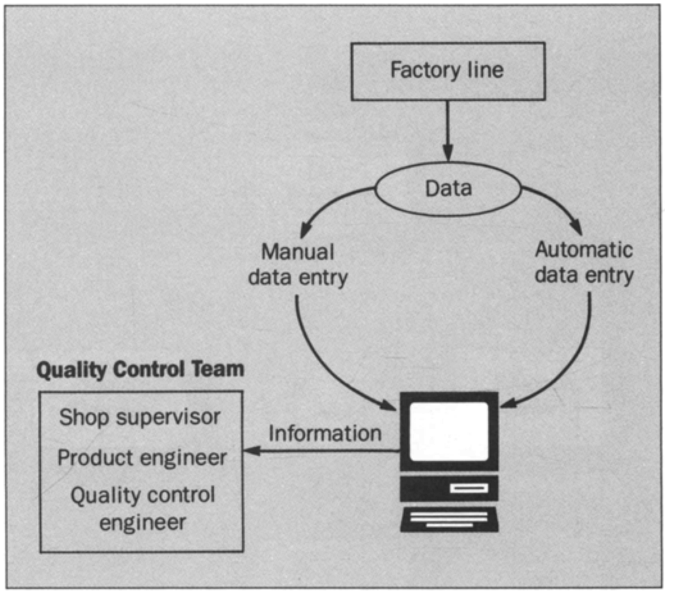


Figure 9. QC Toolkit used as a stand-alone system.

- *Process capability studies.* QC teams can use QC Toolkit to obtain control over a process by detecting, identifying, diagnosing, and removing sources of variability and defects.
- *Process improvement.* QC teams can use QC Toolkit to analyze process variables and determine optimum operating conditions in order to reduce scrap, rework, and defects.
- *Quality reporting.* With QC Toolkit linked to a centralized factory database, QC teams can access data from any shop in the factory and use QC Toolkit to produce reports for management, customers, or vendors.

In general, QC Toolkit can be used to perform quickly and easily those tasks that the process checking system was not designed to do.

QC Toolkit without process checking systems. In factories without process checking systems, the data are entered into PCs. (See Figure 11.) In this case, each shop (or line or cell)

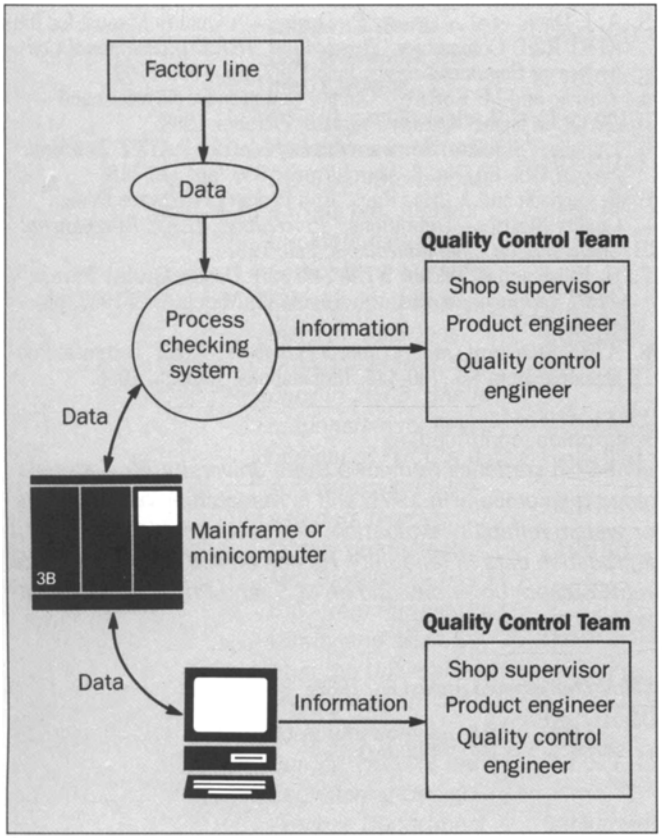


Figure 10. QC Toolkit integrated with a factory information system and a process checking system. The data from the factory line are entered and checked through the process checking system.

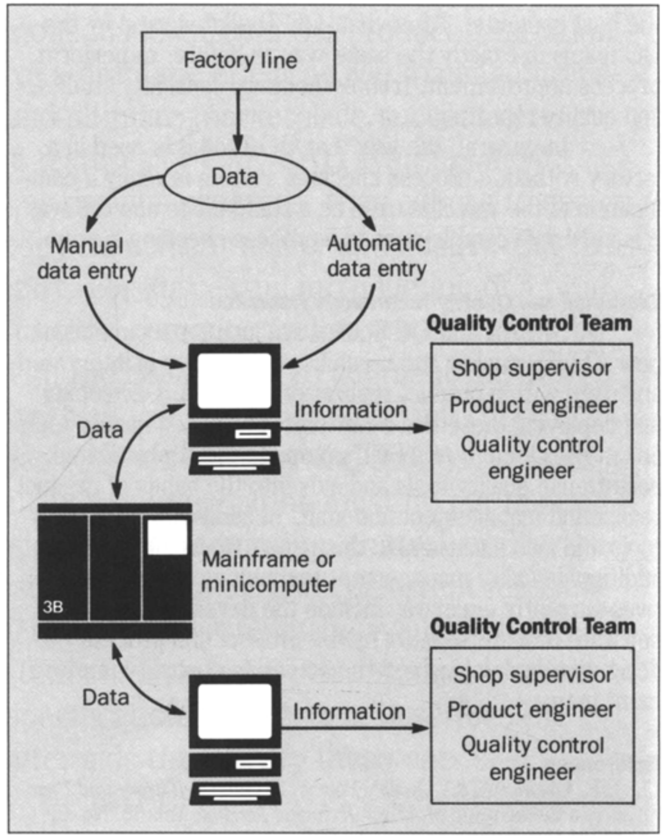


Figure 11. QC Toolkit integrated with a factory information system without a process checking system. The data from the factory line are checked by QC Toolkit and then uploaded and stored on a host computer.

has its own PC. The shop PCs contain both QC Toolkit and a "local" database consisting of a collection of ASCII files. As before, the factory information system is running on a host computer, where the "main" database is kept.

As in the case of using QC Toolkit as a stand-alone, data from each process goes into separate files either through QC Toolkit (manually) or directly into the file (automatically). As soon as the data are entered, they are analyzed by QC Toolkit to detect and flag any out-of-control conditions in the manufacturing process.

As before, shop PCs are connected (via a Datakit network, for example) to the host computer. The shop data stored in the "local" databases on the shop PCs are periodically (daily, perhaps) uploaded into the "main" database on the host computer. After that, QC Toolkit is used by the QC teams in exactly the same way as before: to perform process improvement, troubleshooting, capability studies, and quality reporting.

In general, the way that QC Toolkit is used in a factory without a process checking system is really a combination of the way it is used as a stand-alone and the way it is used as a complement to a process checking system.

Extending the Quality Technology Resource

SUPER and QC Toolkit are just two examples of how AT&T's quality and reliability technology is integrated and used within product realization systems. Developing and deploying these and other software-based quality tools is a major element of AT&T's move to put sophisticated, easy-to-use quality tools and aids into the hands of product realization management and staff. In addition to the many tools and aids discussed in this paper, work continues on additional quality management technology. Major initiatives currently underway include the development of software tools to support robust product and process design methodologies and the activities of quality improvement teams.

References

1. J. E. Olson, "AT&T Quality Policy," in *Quality: Theory and Practice*, a theme issue of *AT&T Technical Journal*, Vol. 65, No. 2, March/April 1986, p. 28.
2. R. D. Meon, "Strategy for Learning and the Deming Cycle," *Philadelphia Area Council for Excellence Roundtable II*, Associates in Process Improvement, Document 16, 1986.
3. A. J. David et al., "Quality By Design—A Quality Manual for the AT&T R&D Community," *Proceedings, IEEE International Conference on Communications*, June 1987, pp. 1766-1769.
4. Y. Akao and M. Kogure, "Quality Function Deployment and CWQC in Japan," *Quality Progress*, October 1983.
5. J. Inglis, "Standard Software Quality Metrics," *AT&T Technical Journal*, Vol. 65, No. 2, March/April 1986, pp. 113-118.
6. E. Anglade and A. Mac Rae, "End Product Hardware Design Quality Metrics—Definitions," *Proceedings, IEEE International Conference on Communications*, June 1987.
7. H. Fallah and G. Tucker, "The Software Design Quality Review," *AT&T Technical Journal*, Vol. 66, No. 2, March/April 1987, pp. 40-49.
8. *AT&T Statistical Quality Control Handbook*, AT&T Technical Publications, Item No. 700-444, Indianapolis, Indiana, 1956.

Biographies (continued)

an M.S. in statistics from Iowa State University. Mr. Tortorella joined the company in 1975 and is researching new methods for system reliability evaluation, and statistical inference for degradation data in reliability. He has an A.B. in mathematics from Fordham University and an M.S. and Ph.D. in mathematics from Purdue University.

(Manuscript received August 28, 1987)