Small-Team Development in a Competitive Environment

James P. Moreland
Eileen M. Sieli
Mary Socratous
Susan C. Udovic

The telecommunications industry is undergoing rapid and continuous change. To be successful in today's dynamic market, competitors must quickly identify and embrace new opportunities. Fortunately, companies have at their disposal enormous amounts of data and high-speed computers that can help them analyze opportunities and hasten their entry into new markets. Unfortunately, the software needed to support these efforts takes precious time to design and develop. Traditional software development methodologies might not suffice in such a fast-paced environment. Thus, alternatives are needed to shorten response time and provide greater flexibility to market-driven changes. This paper discusses AT&T's successes with a small-team approach to software development. Rather than maintaining separate functional organizations having well-defined roles and responsibilities, a single, closely knit team is established consisting of customers, systems engineers, and developers. The team's overarching goal is to help AT&T customers solve business problems as quickly as possible.

Introduction

Today's telecommunications industry is vastly different from that of little more than a decade ago. Competition, new and emerging enterprises like the cable TV industry, as well as cellular technology, increasing demand for innovative services, and the elimination of market-entry barriers all contribute to an exciting and rapidly changing business environment.

To be successful in such a dynamic market, companies must quickly identify and capitalize on new opportunities. Furthermore, what could seem like a good opportunity one day might not be so good the next. The telecommunications marketplace is not only progressing swiftly. It is also frequently and rapidly changing direction.

Fortunately, companies have at their disposal enormous amounts of data and high-speed computers. Such resources can help them analyze new opportunities and potentially achieve immediate market entry. Unfortunately, designing and developing the software to support these efforts takes precious time.

Software developers must continue to find new and faster ways to meet customers' business needs. Development cycles of 18, 12, 6, and sometimes even 1 month may be too long in the rapidly changing telecommunications marketplace.

Traditional development methodology in which system requirements, architecture, development, testing, and implementation are done sequentially by separate organizations—and sometimes even in separate geographical areas—may not be appropriate in today's environment. Alternatives are needed to shorten response time and provide greater flexibility to market-driven changes.

This paper describes some successful implementations of the small-team approach to system development in AT&T.

The responsibilities of separate functional organizations having well-defined roles and responsibilities are instead assumed by a closely knit team consisting of customers, systems engineers, and developers (see Figure 1). The team's principal goal is to help AT&T customers solve their business problems as quickly as possible.

The following two factors are vital to the success of the small-team approach:

- Continually involving customers in all aspects of the development process, especially in graphical user interface (GUI) design; and
- Delivering the product via a number of small releases, each of which provides some immediate customer benefit and allows for early feedback on the overall product.

The entire team understands the customer's business problem, and each team member helps to decide on the optimal solution. The team also understands that the best solution might not even be a software program. Perhaps some paper-and-pencil analysis is really all that is needed or all that time will allow.

This software development approach is by no means new. In fact, it is similar to the Spiral Model of Software Development.[1] Furthermore, many AT&T organizations are implementing methods that significantly improve software productivity and quality by moving away from the traditional "waterfall" model to more flexible development methodologies.[2] Such methodologies contain many more keys to rapid software development—for example, those provided by *object oriented design*, which include the reuse of objects already built for other applications.

A *development environment* much less than a methodology is being described here. This environment is one in which AT&T becomes fully involved with a customer's business. Thus, AT&T partners with its customers in helping to meet the challenges of the constantly changing telecommunications marketplace.

One more purpose of this paper is to encourage other companies to embrace the small-team approach to software development. Many believe strongly that in the future, developers who thoroughly understand the business they support will be successful, at least in the case of information management software systems. This is not just the authors' opinion. Many AT&T customers have stated that they, too, share this conviction.

Panel 1. Abbreviations, Acronyms, and Terms

GUI—graphical user interface
ISO—International Organization for Standardization
LEC—local exchange company
NTD—network technology development
SAFER—Split Access and Flexible Egress Routing
WBS—work breakdown structure

The small-team development approach was successfully applied to the design and development of operations systems that support the capacity management and strategic planning functions for the AT&T network.

The small-team approach was also successfully applied to much larger development efforts, which were segmented into smaller, distinguishable, and more manageable parts. Such parts generally consist of multiple and separable functional applications supported by a common database and a common GUI.

Strategy For Small Teams

AT&T's success with the small-team approach to operations and information systems development is based on the following principles:

- Small teams should not be composed of more than eight people.
- The team must follow standard project management techniques.
- Individual roles and responsibilities must be clearly defined and communicated.
- Team members should contribute outside as well as inside their roles and expertise.
- The team must always include experienced people in project management, systems engineering, and GUI design.

These principles are not unique and can be found in many other software process models. Execution of these basics, however, differentiates the AT&T approach from other models. An approach has been created that not only is risk driven[1] but also focuses on providing business solutions to AT&T customers.

Experience suggests that a small team should constrain its size to no more than eight members. As the number of team members increases, so does the amount

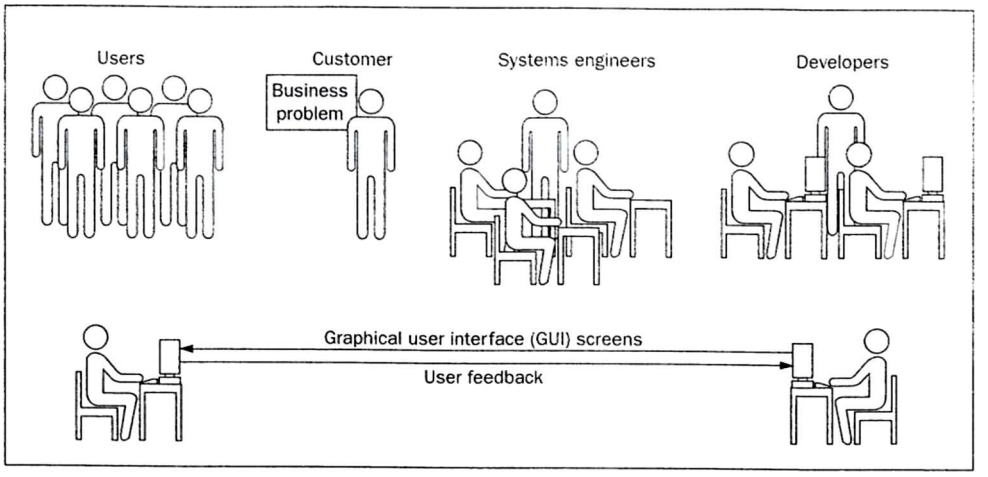

Figure 1. In discussing the small-team approach to software development, the responsibilities of separate functional organizations having well-defined roles and responsibilities are assumed by a closely knit team consisting of end-users, customers, systems engineers, and developers, as shown in the drawing. After initial meetings between the systems engineer, end-users, and customer, the systems engineer meets with developers and the graphical user interface (GUI) designer. Customer comments are then fed back to the team. Such continuous, timely feedback is vital to a more usable software product.

of structure needed to keep everyone informed and organized. Co-location helps to minimize such structure. Teams that are spread across multiple locations face more communication problems. When more structure is needed, both meetings and paperwork increase. Ultimately, software delivery cycle times become longer.

These influences can be counteracted by subdividing large projects into smaller, manageable releases. For example, a large project supporting the AT&T access management price process required 25 new features. Using the small-team approach, the project was segmented into several manageable phases instead of increasing the team's size or subcontracting with outside vendors to complete certain portions. The project also was divided into three incremental releases during the year, and they provided the highest priority features first. AT&T customers were very satisfied with the incremental releases, and this strategy kept the team small and co-located.

This small-team approach stresses the importance of clearly defined roles and responsibilities and reduces conflicts among team members. Managing roles can be quite complex for small teams because roles change as a project evolves to ensure all activities are completed. To avoid confusion and successfully transition roles, each team member must understand the roles to which each individual is assigned. Time is precious, and it cannot be wasted on duplication or unnecessary rework. Experience has shown that if a team member is uncertain about individual responsibilities, something will inevitably "fall through the cracks."

Although one person takes responsibility for a particular role, the AT&T approach often dictates that team members assume additional tasks to complete a delivery. The team develops a work breakdown structure (WBS) to help decide what tasks must be accomplished for a project. The WBS is very useful in identifying potential problems and for ensuring that tasks are not overlooked.

When tasks are identified but not yet assigned, team members are encouraged to volunteer for the roles. Sometimes, this means working outside the area in which an individual is qualified. In such a situation, a delicate balance must be achieved. The ideal is to maintain a high degree of skill in one's area of expertise, but team members must also be willing to help out in other areas. It is important that team members be able and willing to assume responsibility for getting a job done—whatever it takes.

To help ensure success, teams always include an experienced project manager, systems engineer, and GUI designer whose goal is the delivery of a business solution to AT&T customers. These key team members constantly review the project and the team's work. They must be proficient at investigating the issues and finding possible solutions to problems, and they can even commission an outside consultant if necessary. Furthermore, they serve as role models for new, inexperienced team members and actively mentor them.

As previously mentioned, individual team members are expected to contribute inside and outside the roles for which they are qualified, and such contributions also include feedback to other team members. Thus, AT&T's policy with regard to small project teams is that success is enjoyed by everyone. Furthermore, the responsibility for failure must be accepted by the entire group. Some errors are unavoidable, but the responsibility for catching them is shared by all. This approach stresses teamwork, group problem solving, and helping others. When reward is based on the success or failure of the entire group, team accomplishments supersede individual achievements. When team goals and accomplishments prevail, then important issues are less likely to be overlooked.

Small teams must be self-managing with conflicts and decisions being handled democratically. Each member must feel comfortable giving feedback freely to other team members. To facilitate self-management, team members must be peers, and a level of respect must be maintained for each individual's role. With individuals acting as experts in their roles, each member's input becomes invaluable. The team must truly feel that it can rely on each member's contributions.

Small teams cannot accommodate one "star" or a project manager who dictates rather than orchestrates. A feeling of kinship must always prevail among all the members, who share equally in the team's successes and failures. This philosophy benefits the individual as well as the team. Leadership develops naturally in an environment in which each person is given responsibility, is treated as a valued member of the team, and is allowed to contribute freely.

Managing Small Teams for Business Solutions

Unlike many small-team approaches, the incorporation of formal project management techniques is crucial to success. In today's highly competitive, rapidly changing environment, AT&T small projects usually operate under rigorous constraints, including tight schedules, restricted resources, and changing business needs. These constraints leave no margin for error, and they have forced small teams to strive for managerial breakthrough,[3] a term for dynamic movement that results in improved performance.

The AT&T project management approach is built on the standard customer-supplier model,[4] and the company treats all its customers as if they were external. The customer-supplier model provides the structure to establish and maintain customer relationships for all projects. The model emphasizes the importance of both understanding customers' needs and obtaining customer feedback to successfully transform project inputs into the desired set of customer outputs.

The importance of early and frequent customer feedback cannot be overemphasized. It can provide early validation of designs, as well as advance warnings about deliverables that might not meet a customer's needs. Often, project modifications can be made to improve the final outcome without affecting the schedule or resources. Although all team members are encouraged to pursue customer feedback actively, the project manager coordinates it. Such coordination prevents customers from being overwhelmed by the team and from repeating the same feedback to different team members.

AT&T follows the network technology development (NTD) process, an ISO-consistent and AT&T-approved development procedure. The NTD process consists of a defined set of entrance and exit criteria for each phase, and it includes the use of *quality gates*, which are formal process assessment points. By following a defined process, a vehicle was created by which to capture lessons learned. From project to project, what works well can be sustained and the areas exhibiting problems or weaknesses can be improved. The result is continuous improvement, not only to AT&T Bell Laboratories projects but also to customer-delivered products.

Bell Laboratories manages projects end to end, from conception through maintenance. This end-to-end management is an important aspect. It distinguishes the organization's small teams from others that concentrate only on development and that transfer the project else-

where for production or maintenance. Dealing with the same core team for the project's entire life cycle—an "easy to do business with" attribute—is preferred by many AT&T customers and end-users.

Bell Laboratories invests considerably more time than traditional development teams in understanding AT&T customers and their needs. Developers have found that they cannot rush through a project's planning or definition phases. They must fully understand the project's objectives before optimizing its implementation and deliverables. The first step is to meet with a customer to discuss the customer's specific business needs and possible solutions. Developers are included in these meetings so that feasibility issues can be discussed in parallel. Either the project manager or systems engineer attends all customer meetings. Bell Laboratories feels strongly that these two team members must play a lead role in customer relations.

Frequently, GUI screens are prepared during a project's definition phase to illustrate the planned approach. These screens are reviewed with a customer to obtain early feedback. This activity also helps define constraints, identify potential risks, and confirm assumptions. Managing these factors becomes easier when each is identified early in a project's life cycle.

When the requirements are confirmed, the project manager meets with the team to discuss further the feasibility of each feature, then negotiates the solution's scope or the set of deliverables with the customer. It was found that this approach steers small teams on the right course to customer satisfaction by providing the flexibility to react to changing needs. Progress is ensured by always targeting a customer's top priorities.

The Project Manager's Role

As previously mentioned, AT&T's small teams are self managing and democratically oriented, and a project manager ultimately directs the team. This individual, however, cannot fulfill all the responsibilities of the position without being accepted by each team member. Control must be given, similar to the concept of *right of way* when driving a car. One can never take the right of way—it can only be given. Likewise, the project manager—as a peer—cannot take control of the project's organization and direction. Such control must be granted by the rest of the team.

For a project manager, *customer liaison* is the most important responsibility. A project completed on time and within budget is still not considered a success if the customer is not satisfied. Understanding customer needs and priorities, as well as managing expectations, must all be completed professionally. As previously mentioned, most team members frequently interact with the customer. It is the project manager, however, who assumes the responsibility of managing customer interactions for the project's duration.

A project manager also has other responsibilities. This key individual ensures the continuous exchange of information among all those associated with a particular project. To promote communication within a small team, a project manager conducts weekly status meetings. Furthermore, the manager monitors every aspect of a project, overseeing not only the schedule but also such crucial aspects as changes in customer priorities, problem resolution, and the provision of required inputs. The project manager tracks the project from beginning to end to assess the project's status and to provide input for future development estimates.

The Bell Laboratories tailored project management approach focuses continually on understanding customers in their work environment and the requirements placed on them. Frequent customer feedback is solicited to assess the project in terms of meeting customers' top priorities. Furthermore, the importance of the team itself is not forgotten. It is the dedication of the individual team members who work together as one that ensures the project's ultimate success and customer satisfaction.

The Systems Engineering Role

Bell Laboratories' role as systems engineer is fourfold: to understand a customer's business problem; to formulate an appropriate solution; to produce requirements for development; and to ensure that the development team fully understands the business problem.

Key to success in the systems engineer role is understanding a customer's business function. Experience has shown that the best way to acquire the proper training is to meet with customers and end-users and understand how they perform their jobs. If other processes receive inputs and provide outputs to and from a customer's business function, a familiarity with those areas is required as well.

Besides business knowledge, a good under-

standing of a customer's process and system needs is also key to the success of the project. It is important to understand what users are trying to achieve within their process. The systems engineer also needs to be aware of any existing systems outside a customer's process that could be reused to meet some of the needs. For example, in a previous project, the team needed to match access tariff rates filed by the local exchange companies (LECs) to LEC facility counts on access bills for special services to calculate access expense. A system already existed, however, that matched the tariff rates to access volumes from the bills. Thus, the team only had to use system outputs in calculating access expenses.

After understanding the problem being addressed, providing the best solution is the next priority. The first step is to consider when a customer's process deliverables are due. Often, a crucial date exists, and it is typically associated with a business problem. If a solution to a customer's problem arrives after the crucial date, then the solution is of no use. The best solution may require building a new system or function, or it may involve streamlining a user's process using existing systems or providing a one-time analysis.

Proposed solutions must be reviewed with both the customer and end-users. Experience highlights the importance of maintaining customer and end-user involvement in the process of defining proposed solutions. Customers prefer this approach rather than being excluded from the definition of their systems. The success of the final product is ensured when customers and end-users remain involved throughout the entire development process. In a recent project, the development team collaborated with an experienced end-user who reviewed the team's approach in meeting the needs of each software feature built. The end-user was also actively involved in designing the user interface for the tool.

When everyone concurs on the approach to a problem, the solution is then subdivided into small functions that a customer can prioritize. This approach has also been applied to larger projects so that small development teams can be assigned to work on individual functions.

A small team's systems engineer must also assume a combination of roles as required to fill any resource gaps in producing a deliverable. Furthermore, each of the other team members might have to perform many tasks to meet a customer's needs.

A small team's systems engineer need not communicate as formally with other team members as is customary in a large team. All new requirements and any changes to those already existing, however, still must be formally documented. And yet, team meetings must be held to ensure that requirements are understood. If the process is compromised—for example, by not documenting every single requirement and relying instead only on word of mouth, then product quality is jeopardized. Because there is less formality within a small team, the process is more flexible and can respond faster to changes in requirements.

Development

Within the small-team environment, development requires the same organization and process control as that for large teams. At times, it has been tempting to streamline development because the team is small. It has been proven, however, that the needs for designing for reuse, efficient code documentation, and interface agreements do not diminish with fewer developers. For example, designing for reuse is even more important when cycle times are short. Furthermore, the development effort must be clearly divided because the team has even less tolerance for confusion about who is doing what. The small-team development process works best when thought of as a microcosm of the large-team development process.

The small-team process involves iterating through the requirements collection, design, development, and customer review. After initial meetings between the systems engineer and customer, the systems engineer meets with the GUI designer, who then produces a document that developers use to define a phased approach. Customers also use the document to review features as they are developed. Customer comments are then fed back to the team to ensure that the project is headed in the right direction (see Figure 1). Such continuous feedback is vital because the sooner customers review the team's progress, the fewer changes that will be needed and—ultimately—the more usable the product.

A GUI designer collaborates closely with a systems engineer to transform customer needs into a work flow through the system. These customer needs are then used as the GUI design. The GUI designer works halfway

between the systems engineering and development domains, defining screens at a high level. Most low-level GUI decisions are either enforced by the development environment or left up to individual developers. It is the responsibility of the GUI designer, however, to ensure consistency and usability.

High-level application development environments like PowerBuilder* and Visual Basic* are used. These tools provide software components or objects that can be used as building blocks. These building blocks reduce development time, encourage reuse, and facilitate modifications. Being able to modify code quickly or create a demonstration in several days is essential. The tools also eliminate much of the low-level decisions about GUI design (for example, designing yet another database query screen). They free the GUI designer to concentrate on higher-level design issues. Many tools provide predesigned screens or encourage a consistent way of using components. These options provide standards and guidelines for many low-level GUI design issues.

Source code and version control are musts, even for teams on which only one developer is present. A software architecture that enables reuse by identifying and building generic functions is encouraged. A systematic method is also provided for recording end-users' modification requests.

The short development schedule does not permit a large deliverable. Much negotiating with the customer is necessary because features must be prioritized and delivery done incrementally. The systems engineer is relied on to interpret and prioritize customer needs, and the project manager negotiates deliverables based on time estimates from developers.

Often, with a non-negotiable delivery date, an unexpected need arises. This situation sometimes necessitates postponing scheduled features and attending to the immediate need. The team must be flexible enough (and willing) to accommodate such postponement. At times, the software solution was even abandoned. In one instance, it was physically impossible to deliver software to replace end-users' current spreadsheet program by the year-end deadline. Instead, an agreement was reached to obtain and format data to facilitate the current process, enabling end-users to deliver to their customers on time. The team then negotiated to deliver the new software later. Development was suspended for two days until the data feed could be prepared. The objective is either to meet the customer's immediate need or to find someone who can, not to be a "development shop."

Even though all team members conduct testing and provide feedback, the project manager, systems engineer, and GUI designer double as system testers and are ultimately responsible for it. Some customers also conduct system testing. Such an arrangement facilitates faster delivery because both system testing and customer acceptance testing are combined. In addition, customers test the system in ways the team cannot anticipate.

End-users install their own software by executing an installation program across the network from a server containing the small team's product, as well as other products from organizations producing software for the same platform. These programs, commissioned by a common end-user community, access shared databases remotely within a client-server platform and appear as icons in the same program group in the Microsoft* Windows* environment. New versions are announced by means of e-mail, and users are encouraged to upgrade as soon as possible. Servicing a large user community with small, modular software products produced by independent but coordinated small teams is facilitated by the client-server environment.

Technical support is provided through the team. Such provisioning gives team members the opportunity to talk directly with end-users about specific problems and assists in continually refining the system.

Sometimes, like all software developers, the team may receive urgent change requests resulting from misunderstanding end-users' needs. Other times, feedback can consist of, "Yes, I know what I said, but now that I see it, it's not what I want." Customers might be defining a new process through the software solution being provided, preventing them from envisioning their needs in advance. While the luxury of prototyping cannot be afforded, the process has many built-in advantages to facilitate dealing with such inevitable problems.

Because the team works so closely with customers, many software demonstrations are conducted during development. Formality is not a major small-team issue. Thus, customers are permitted to review even unfinished, error-prone software. In addition, the simple fact that the team delivers fewer features in a shorter time allows it to provide customers with a "taste" of the

not yet completed product. Then, it can modify and redesign the software if necessary.

Another advantage of this approach is having customers participate in system testing. Such participation provides yet another chance to catch errors in design.

Finally, if a problem remains undiscovered until production, changes can still be made quickly, and the software can be delivered as fast as the customer can install it.

Applications

The small-team development approach has been successfully applied to planning and implementation systems that support the access network—that is, the network that connects the AT&T public switched telephone network with those of the LECs. Such systems have allowed AT&T to achieve substantial reductions in access, labor, and capital costs. For example, the AT&T Access Planning and Grooming System identified more than $200,000 in savings in access charges by economically routing and bundling access circuit demands to better use DS1 and DS3 facilities. AT&T customers using the Access Planning and Grooming System credited AT&T with setting a record for the development of a system of this size.

Furthermore, the AT&T Servicing, Ordering, Forecasting, and Expert Advisor System is well on its way to automating the capacity management of the AT&T switched access network. AT&T customers have said they cannot believe so much was accomplished so quickly, and that they specifically like the concept of AT&T people working with a team of users ... to "blue sky" an idea ... build it ... improve it ... .

The AT&T team also developed the Split Access and Flexible Egress Routing (SAFER) access reliability product and the SAFER benefit tool, and the team worked directly with other major corporations to demonstrate AT&T's leadership in network reliability. The team completed switch requirements, software development and testing, network provisioning and maintenance procedures, and pricing all in less than one year's time, allowing AT&T Business Communications Services to meet an urgent marketplace need quickly.

An 800 Services Visualization tool was developed to help AT&T account managers sell advanced 800 service features. An AT&T Consumer Communications Services graphical analysis tool helped analyze the effects of the business unit's promotional campaigns. Both tools, which received high customer praise, were developed within a few months to help AT&T customers meet narrow market windows.

The team is currently developing a *customer location database* to support the introduction of competitive forms of access. Customers particularly like the small-team approach because it allows them to be flexible and to respond quickly to changing customer needs.

The team is also developing a *supplier tariff analysis resource tool* to help automate, standardize, and improve the quality of the AT&T Access Price Process. AT&T customers have said that the team was extremely responsive to process requirements. They stated further that it proactively anticipated, sought, and participated in the development of customer-specific system and software solutions. Customers not only are pleased with the team's intent and dedication to meeting established schedules. They also appreciate the team's flexibility in helping to meet new and changing business needs.

Summary

The AT&T small-team approach to helping customers solve business problems has proven to be successful over a wide range of projects, and it has been well received by many AT&T customers. This success was achieved by focusing continually on understanding customers in their work environment and by frequently soliciting feedback to ensure that customers' top priorities are being met.

*Trademarks

Microsoft is a registered trademark and Windows and Visual Basic are trademarks of the Microsoft Corporation.

PowerBuilder is a registered trademark of the Powersoft Corporation.

References

1. B. W. Boehm, "A Spiral Model of Software Development and Enhancement" *IEEE Computer*, IEEE Publishing Company, May 1988.
2. S. Chen, "Developing AT&T's World-Class Competence in Software," *AT&T Middleware Day and Software Symposium*, Holmdel, New Jersey, October 1995.

3. J. M. Juran, *Managerial Breakthrough*, McGraw-Hill Publishing Company, New York City, 1964.
4. E. M. Sieli, "Managing a Project as a Process," *AT&T Technical Journal*, Vol. 70, No.2, March/April 1991, pp. 33-39.

(Manuscript approved January 1996)

James P. Moreland *is a technical manager for the Customer Connectivity Planning and Development District in the Network Management, Capacity, and Access Systems Division at AT&T Bell Laboratories, Middletown, New Jersey. He is responsible for the design of AT&T access and local networks, as well as the development of software systems for these networks' performance and financial management. Mr. Moreland has B.S., M.S., and Ph.D. degrees in electrical engineering from Ohio State University in Columbus. He joined AT&T in 1968.*

Eileen M. Sieli *is a distinguished member of technical staff in the Network Management, Capacity, and Access Systems Division at AT&T Bell Laboratories in Middletown, New Jersey. She is responsible for the project management of small teams that provide software tools supporting AT&T's business strategies for realizing access cost reductions. Ms. Sieli has a B.A. degree in physics from Adelphi University in Garden City, New York, and an M.S. in electrical engineering and computer technology from the New Jersey Institute of Technology in Newark. She joined AT&T in 1985.*

Mary Socratous *is a member of technical staff in the Network Management, Capacity, and Access Systems Division at AT&T Bell Laboratories in Middletown, New Jersey. She is responsible for systems engineering and database design within a small development team. Ms. Socratous has a B.S. degree in electrical engineering and an M.S. in computer science. She joined AT&T in 1984.*

Susan C. Udovic *is a member of technical staff in the Network Management, Capacity, and Access Systems Division at AT&T Bell Laboratories in Middletown, New Jersey. She is currently developing software and graphical user interfaces (GUIs) for the AT&T Network Services Division's Customer Connectivity Planning and Development Group. Ms. Udovic has a B.A. degree from Douglass College in New Brunswick, New Jersey, and an M.S. from Rutgers University in Piscataway, New Jersey—both in psychology. She joined AT&T in 1984.*