Iterative Design and Implementation
A Model of Successful Scheduled Maintenance Technology Deployment

E. STERLING CHAMBERLAIN III AND LIA A. DI BELLO

With significant budget reductions, many transit providers are finding that they must examine the way in which they maintain their fleets. As the use of outsourcing to stay within budget increases, those serious about maintaining their own fleets are looking more closely at totally new approaches that emphasize the efficient use of funds and labor resources with the goal of maintaining high fleet performance. Preventive maintenance has been accepted as a solution, and many transit industries are investigating computerized decision support systems (DSSs), often borrowing technologies developed for related industries (e.g., aviation) and tailoring them for transit purposes. Unfortunately, most implementations fall short of expectations. To be effective, DSSs require users with an in-depth understanding of their logic and detailed knowledge of the work being done. Mechanics and foremen have the work knowledge but are unlikely to possess an understanding of computerization and the abstract algorithms driving these technologies. One system adapted to the transit industry by New York City Transit—MIDAS—has been implemented successfully in the Department of Buses. The three-part top-down/bottom-up model for technology deployment is described, as is the evaluation used to measure its success: a relatively small design team of software developers, maintenance managers, shop-floor workers, and foremen; a powerful new training technique resulting in direct data entry from the shop floor; and feedback from the work floor to tailor the system. It is concluded that deployment must be organized around the concept that DSSs are tools appropriated by the users and that implementation efforts should facilitate that process. This is in striking contrast to "turnkey" approaches. Follow-up evaluation data are presented that show how this model leads to more innovative and effective system use over time.

With significant budget reductions, many transit providers are finding that they must examine the way in which they maintain their fleets. As the use of outsourcing to stay within budget increases, those serious about maintaining their own fleets are looking more closely at totally new approaches that emphasize the efficient use of funds and labor resources with the goal of maintaining high fleet performance. Economic functioning has always been a goal in maintenance, but aggressive strategies were not typically explored in the public transit industry for many reasons, most of which are well known.

The first of these reasons is the way that public service budgets are developed and maintained. In the area of transit maintenance they are typically built by "backing into" a total dollar amount that is some version of last year's budget. This approach ensures that "last year's" maintenance will produce "this year's" budget. Therefore, there is little incentive for managers to develop and maintain efficient uses of funds; if they come in under budget in one year, their future operating budget may be decreased and the excess shifted to less efficient operations. In other words, innovative managers are not given the option to reinvest their savings and produce a higher-quality product, and they often see their savings passed on to others.

Second, serious disagreement exists over the best ways to reduce costs while providing the most service. In a sense, continuous service and assiduous maintenance are contradictory activities, as maintenance of any kind requires vehicle down time. New York City Transit (NYCT) has tried to manage this contradiction by using scheduled inspections based on various parameters, such as calendar days or mileage. However, this approach identifies parts only after they are visibly defective. At bottom, scheduled inspections of all kinds produce reactive maintenance and can never be truly preventive.

Unfortunately, at this point, nearly all maintenance policy is guided more by politics and folk wisdom than by any detailed data collected on vehicles in service. Such practices have appeared necessary—or are accepted as a good second best—as efforts to collect and maintain usable data on large fleets have proved impossible using traditional paper records. However, as transit providers are under increasing pressure to produce strategies with real financial impact, folk wisdom is subject to greater scrutiny.

With advances in information technology, preventive maintenance has been increasingly considered as a potential solution. Many transit industries are investigating computerized scheduled maintenance decision support systems (DSSs), often borrowing technologies developed for related industries (e.g., aviation) and tailoring them for transit purposes.

The idea behind scheduled maintenance technologies is that, with sufficiently detailed data collection and analysis, the failure rates or life cycles of subassemblies and components can be predicted in vehicles exposed to relatively constant conditions (urban versus rural use, for example) and cost-effective preventive replacement of parts can keep vehicles in service. Scheduled part replacement would thus replace other time-consuming activities, such as inspections for failure, vehicle breakdowns, and waiting for parts that may not be available for these unplanned needs.

A scheduled maintenance system is a powerful tool for collecting the data required to identify component life cycles and to schedule vehicles for replacement actions. These life cycles can be expressed as the number of days or miles (or other variable) of useful life for each component. A table of life cycles can then be developed and used to create template work orders (work orders that generate themselves according to cycle-based rules). The template work order will then prompt shop-floor personnel that a particular part on a particular bus is due to fail and ought to be

---

E. S. Chamberlain III, New York City Transit, Department of Buses, 41–36 51st Street, Woodside, N.Y. 11377. L. A. Di Bello, Workplace Technology Research Group, Center for Advanced Studies in Education, City University Graduate School, Suite 620, 25 West 43rd Street, New York, N.Y. 10036.

replaced (Table 1). Using the life-cycle tables developed with the data collected by a DSS, vehicle maintenance can be planned by coordinating the life cycle–driven component replacements. A vehicle can then be scheduled for maintenance at a point that includes several component changes at a time close to the optimum life cycle for each item.

Unfortunately, most implementations fall short of expectations. Typically, two areas are identified in relationship to technology implementation failure. The first is the user: to be effective, DSSs require users with an in-depth understanding of their logic and detailed knowledge of the work being done. Mechanics and foremen have the work knowledge but are unlikely to possess an understanding of computerization and the abstract algorithms driving these technologies. In addition, many such individuals resist the idea that decision support technologies are part of their jobs. In contrast, mechanical engineers—the usual target users from a vendor's point of view—often do not possess the in-depth knowledge of processes to finely tune these systems (*1*).

The second area of difficulty is the system itself and its distribution or placement in the workplace. Off-the-shelf systems frequently are not tailored to work practices or maintenance priorities and appear irrelevant or awkward to the user. In addition, little thought is usually put into where the technology should go. For example, protecting the terminal or the CPU from theft or dirt often becomes a higher priority than convenient access for the target user.

One system adapted to the transit industry by NYCT—MIDAS—has been implemented successfully in the Department of Buses. This paper will detail the three-part top-down/bottom-up model for technology deployment and the evaluation used to measure its success. As will become clear, this model differs radically from typical technology implementation, and does so, in part, by making the usual cause of technology failure—users' resistance surrounding the system—part of the solution.

MIDAS PROJECT

History and Formative Research

Late in the 1980s NYCT began investigating maintenance information. A committee was formed to examine previous attempts to implement similar programs and the factors contributing to their failure or success. This analysis revealed that any new system must be work order–based, must clearly establish a link between defects and repairs, and must give a real-time picture of the status of buses under repair. More important, the analysis showed that data entry must be done by those who could actually detail repair activities—ideally, mechanics themselves—and thus required a user who understood both maintenance operations and the long-term analytic goals of the preventive maintenance system.

The committee set the following criteria for a possible maintenance information system:

- A link between the work order system and the materials management system that would provide data for failure analysis. The idea was to measure part life cycles and rate of use.
- Unit history for all rebuildable units to track all units installed on an individual bus and to track the life cycle of a unit across many buses.
- Automated fuel station, collecting fuel, fluid, and mileage.
- Ad-hoc queries that are simple to use.
- Simple procedures for maintenance to check operator defect cards and to establish work orders for repair.
- System should be able to track the status of campaigns.
- System should show information from accident brief reports on line and be able to link the report to the repair work order.
- System should be used by workers who can detail actual repair activities and can also understand how the system will use the data entered.

During a 1-year study of existing maintenance systems, the committee systematically surveyed or visited the 25 largest U.S. transit providers—as ranked by the American Public Transportation Association—to get a sense of what others in the industry had in a maintenance information system. We were interested in whether or not they used a computer system, and if so, whether it was

- Off-the-shelf or custom programming,
- Work order based,
- Capable of predicting component failures,
- Capable of linking repair actions to the reported defect, and
- Based on codes.

In addition, the committee wanted to know in what manner the data were entered (manually or automatically) and who was required to do the data entry (e.g., the mechanics themselves or special staff).

The results of the survey indicated that while most providers had some type of computer information system, they were generally

TABLE 1 Optimum Component Life Cycles Developed from Repair History

	LIFE CYCLE PERIOD IN DAYS	COMPONENT
1	300	BRAKE RELINE
2	180	AIR DRYER REBUILD
3	280	AIR COMPRESSOR GOVERNOR
4	556	FAN BELT
5	556	FAN BELT TENSIONER
6	260	STARTER RELAY VALVE
7	275	DOOR CONTROL MICRO SWITCHES
8	280	FLASHER UNIT
9	550	DRIVER'S SIDE WIPER MOTOR
10	605	FRONT LEVELING VALVES
11	560	DIRECTIONAL FOOT SWITCHES

limited to recording reported defects and tracking safety inspections. No organization had a system that was proactive (i.e., that collected data that could be analyzed to identify mean distance between failures of specific components). In fact, the committee also found that the notions of "preventive replacement" or "preventive repair" were largely misunderstood. Most properties believed that they were accomplishing the same thing by regularly scheduled inspections of the whole vehicle, without regard to repair histories. This attitude assumes that breakdown can be prevented by watching for it. In contrast, NYCT's analysis of actual data showed that regular inspections serve mostly to prevent unsafe vehicles from going into service only if the problem is advanced enough for visual identification.

Providers that had an automated fueling system usually linked it to their information system in order to record fuel and fluid use and mileage. The typical system was not based on a work order, required data input by managers or supervisors, and did not capture specific repair actions or defects. The committee saw very soon that the key element missing in most information systems was a record of specific repair actions for defects. This missing information was the data needed for establishing a true preventive maintenance program as it had been defined.

Three-Part Deployment Model

The extensive research that informed the early stages of the MIDAS project also indicated that—for whatever reason—technology implementation practices appeared as limited as the technologies currently in use. For the MIDAS implementation, three clear goals forced NYCT to rethink and redevelop the deployment model. The first of these was a sensitive system that meaningfully represented the work being done on the vehicles. The second was accurate data. Third and most important, NYCT wanted the system to be actually used; like many providers, NYCT had a history of implementing large technologies that were never accepted or were improperly used. The organization had even implemented a scheduled maintenance system before that was remembered, by the users, as a legendary waste of money and time. The solution used for MIDAS was to involve the end users in the initial design and to allow users at each site to make small, critical modifications during the actual implementation. This required an iterative approach to the final rollout.

Iterative Design and Implementation Process

In the past, new technologies such as a DSS have been designed and implemented by top management. They tend to have been designed by an information systems (IS) department or vendor, and typically the application design is frozen before it is ever seen by the user. The record of success for systems implemented under this model has not been good. For MIDAS the team decided to use a top-down/bottom-up model that would encourage changes in the application design during the early phases of the implementation and allow each new site to have input to the way MIDAS functioned in "their" depot. This approach takes seriously the notion that any system that globally affects work must also map onto what is actually being done and, more important, that one may not know the details of such a system without user input. The problem was, how to get user input before MIDAS was in place and overcome the "not invented here" resistance expected in a multiple-site implementation? The solution was an iterative design and implementation process. Each aspect of this design is discussed in detail in the following sections.

Briefly, the system was first designed in the basic sense (with the vendor) using a small design team of maintenance and operations staff. Next, an in-depth assessment of each site was done by university researchers to measure various aspects of how the depot was functioning and how the workers were understanding and thinking about maintenance operations. This assessment includes cognitive assessments, work flow and artifact analyses, and financial performance measures. These data were collected for two purposes: first, they allowed the university staff to design a tailor-made educational program for the workers, and second, they provided baseline data for later comparison. After the assessment, maintenance workers at all levels were given an aggressive educational intervention (designed using the preceding assessment) that introduced them to the deeper logic of scheduled maintenance theory and systems. In most locations, these interventions involved as many as 150 managers, supervisors, hourly workers, and cleaners. The goal was to enable mechanics and line supervisors to do all data entry and analyses.

During the training, the hardware and software installation took place. Once all personnel had gone through the workshops, the system went "live" in their location and a transition team was placed in the depot until the staff had reached independence with MIDAS. Finally, after a number of weeks, the university researchers returned and conducted—again—assessments at various levels to measure change. In addition, they collected user complaints and suggestions, as did the transition staff. Design changes were made accordingly and added as enchancements to the MIDAS program.

The users and the deployment were assessed again after several months and further design changes were made. The net result of this approach was a finely tuned system that underwent about 200 user-initiated design changes after being implemented in only two sites. It is expected that the number of changes will decrease dramatically in each new site. It is significant that actual use by educated and involved users identified the finer points of the design at individual locations. This allows personnel at each site to feel that they designed MIDAS themselves.

Elements of Deployment Model

Design Team

To tailor the design of the final MIDAS system, the actual end user was included from the very beginning. The design team included experienced operating managers, first-line supervisors, and shop-floor mechanics in addition to the usual complement of IS representatives. The group was facilitated by some experienced project managers from the prime contract holder and the developers of the software package. The design team met weekly for 4 months and discussed each screen of the application with the developers and outlined how the user would navigate through the system. These designs were applied to the program and put into a test system for further review by the team. This somewhat unusual approach to system design was made more so in that the NYCT design team was chaired by a maintenance manager with full authority to make final decisions. It is vital that end users are in real control of the design process. Constraints of the information technology department must be justified to the operating users if a conflict is created by design goals. Although a great deal of support is required to educate the operating members, this type of small diverse team is the most effective structure for developing an application that fits the end user and avoids major obstacles to acceptance.

Education Process

At the onset of the design process, senior management of the Department of Buses decided to have the hourly bus maintainers, Class B (BMBs) enter their own repair data into the system without clerical assistance. This was done to ensure that MIDAS recorded bus maintenance history in the most detailed way possible and moved data entry responsibility directly to the shop floor. The idea is that all repairs would be described by the person most familiar with the actual work: the BMB who did it. There is no "interpretation" of an expert mechanics language by a non-expert data entry clerk. On the other hand, although there is much evidence to show that clerical data entry results in significant degradation of the data, there are also many examples of the difficulties involved in getting shop-floor mechanics to do this type of data entry or to translate their knowledge into the kind of explicit form needed for a computer system. Since this had never been tried before on such a large scale with a "management" technology, the MIDAS team believed that any traditional education approach based on this very different kind of user would be inappropriate at best. Di Bello had established a record at NYCT's car equipment department for overcoming the typical long learning curve and designing educational interventions that appeared to bypass the need for a "prerequisite" knowledge of computers. On the basis of a program of studies that identified what factors of on-the-job experience led to in-depth understanding, Di Bello and her colleagues designed a series of hands-on simulations that basically compress the "incidental" learning that takes place on the job (over a period of a year or two) into a few days.

NYCT was also interested in her methods because the workshops have been shown to mitigate resistance. According to DiBello (*2–4*) this may occur because the real basis for resistance may be more cognitive than attitudinal. That is, numerous studies show that experienced workers possess expertise that makes it possible for efficient intuitive problem solving (*5*). Unfortunately, the same intuitive clarity that experts experience in their domains may also make it difficult for them to see new possibilities. Di Bello's methods are designed to work with intuitive expertise while redirecting it.

Rather than prepackaged workshops, Di Bello and her colleagues design simulations that map onto both the historical work practices and the targeted changes in any given site. Based on this record, the implementation team asked Di Bello and Kindred of the City University of New York (CUNY) to design an intervention for the MIDAS project. They began the process of design with 3 months of assessment at NYCT sites (see description later in this paper). After identifying the trouble spots and developing a user profile, they presented NYCT with the workshop design described here.

The workshop contains three basic components that have proven to be a powerful combination for effectively learning:

1. They are hands-on exercises: learners develop new strategies by solving actual problems in miniature versions of their work environments and encounter the actual results of their decisions.
2. Before being taught new principles, learners are helped (via the activities) to become aware of their "default" strategies and given tools for overcoming them or incorporating them into new strategies. The idea here is to get conscious awareness of automatic problem-solving strategies and integrate relevant portions with the learning of the new system
3. Participants learn the logic of new technologies not by operating the computer system, but by operating *as* the computer system, doing its logical operations manually and constructing the necessary data structures. Only after being the computer do users get to *use* the computer.

For the MIDAS system, CUNY built a three-module workshop. Module 1 is designed to "engage the default." Rather than anticipate the entry point of learners, CUNY designed a workshop in which participants may construct their own entry. Rather than lecture, the workshop trainers give participants a problem to solve with goals that are compatible with the technology—in this case, MIDAS—that they are trying to learn. The trainers also give the participants a number of tools for solving the problem.

For the MIDAS workshop, teams of eight participants were asked to "run a depot" of 40 plastic buses with relatively complex interior components. The goals were to maintain 32 buses in service at all times (limiting the number out of service to eight), order all the materials (within a budget) needed for doing so, and evaluate daily operator reports (each "day" being 20 min) that might indicate potential problems (e.g., noisy engine). The activity was rigged so that the only way to meet these goals was to predict what was due to break next. The breakdown patterns of all components followed time/mileage cycle rules and were precalculated using a computer. The participants were given adequate tools to predict and calculate this breakdown, (printouts of every bus's repair history, among other things) but they were given other tools as well, including those similar to those used to do "reactive" maintenance.

The ways in which the participants used the tools closely approximated their work history. Even when facing a new problem, people tend to construct a solution that fits with their experience, even when explicitly instructed to avoid doing so. In fact, the participants are rarely aware that they are replicating their normal methods. Instead of interfering with this tendency, the trainers allow the participants to "wing it," while carefully documenting the cash flow, labor flow, inventory acquisitions, and the number and type of on-the-road failures that result from failing to predict problems. Meanwhile, heavy fines are levied for expensive reactive problem-solving strategies, such as cannibalizing an entire bus for a few cheap parts that will get other buses back on the road. Later the participants are shown the consequences of their decision-making patterns and the underlying logic used. By the end of the first module, the "depot" is in crisis and the participants are realizing that their budget is being spent to react to mounting problems. The activities are stopped and the team is sent to lunch.

In Module 2 of the workshop, participants reflect on what they did, as recorded by the trainers. The participants discuss among themselves what thinking led to various decisions and begin to identify practices that lead to bad outcomes instead of practices that are preventive. It is only at this point that the participants are truly open to new ideas about how to solve the problems of vehicle maintenance. They also begin to understand in detail the ways that their gut decisions reveal how they have actually misunderstood preventive maintenance. In the last part of Module 2, the participants are facilitated by the trainers in building a manual scheduled maintenance system. The participants identify cyclical patterns from histories (which were available from the first but which now take on new meaning) and set up the predictive data structures, identifying true cycles, and—most important—coordinating cycles so that their "system" is bringing in a bus only once to satisfy several cycles at once. For example, in the MIDAS workshop, the participants quickly realize that a 15,000-mi cycle and a 30,000-mi cycle can be coordinated so that at least half the time the 15,000-mi cycle co-occurs with a 30,000-mi component cycle. The participants then enter these data on an actual test region in MIDAS and create and assign the work orders according to this schedule.

During Module 3, the participants complete their data entry and print out their work assignment sheets and work orders. They run their miniature depot again using MIDAS and see the difference in profits and ease of work flow. Usually only after 5 "days" can the team afford to buy an additional bus to add to the fleet and thereby increase their farebox revenue.

The last activity of the workshop involves entering the data on work orders (paying attention to detailing the components, defects, and symptoms involved) and closing out work orders and work assignment sheets. At this point, participants also learn how to get various reports that they now realize they will want, such as a 30-day history on a bus. After operating *as* MIDAS and then *with* MIDAS, participants navigate through the actual system more easily, know what to look for, and ask informed questions. Even computer-illiterate individuals show little hesitation when exploring the system.

Di Bello and Kindred believe that the workshop's power lies in the first two modules, when the participants actively encounter the basis for their resistance—existing expertise and automatic ways of doing things—and revise it to fit new demands and priorities (*6*). The process of doing this is critical because each individual has a different prior perspective that must be reckoned with. The research informing this method has shown that it is best to facilitate this process of "retooling" one's way of thinking rather than try to overlay new knowledge on top of the old with lecturing. It appears that learning always involves building upon, or reorganizing the way that one already understands something (*1*). Therefore, it is critical to engage prior knowledge, if only to eventually make sure that it is changed or reorganized. Why is it necessary to let the participants do this on their own instead of asking trainers to instruct differently? Research shows that learners of diverse work history backgrounds "enter" into understanding by multiple (*2*). For example, whenever a teacher simplifies material for his or her students, he or she is really anticipating the entry point of the learners. This method often fails with experienced workers because the entry point is not always predictable or universal (for example, "simplified" is often not helpful for those experienced in thinking through vast amounts of detail). The learners—when allowed—actually do better at breaking it down for themselves in a way that is useful to them.

Modification of MIDAS

From previous experience and from discussions with other transit agencies, the MIDAS team recognized that implementing a "final" design of the MIDAS system would be very difficult for two reasons. First, it is hard to fine tune a system's content before it is in use, on-site. Second, having 23 separate and diverse maintenance shops and depots involved introduces some inherent difficulties; each new location would react to MIDAS with "it won't work here" and would feel that the unique qualities of "their" shop had not been accounted for. As a solution, the MIDAS team decided to seek out suggestions and requests from the users at each location as the MIDAS system was implemented and to add these enhancements before the transition was complete. The technical structure to support this level of change must be available in the basic application and is a very important consideration in the selection of a system. In MIDAS most processes are controlled by tables that are maintained by the user from within the MIDAS program. User requests for a different maintenance cycle, a more specific description, or a new repair or component can be met easily by a transition support member at the location. A new report or a search requires some programming, by either the developers or in-house staff, but are added in most cases as requested. The relationship with the vendor must allow for such programming changes.

EVALUATION

In addition to the training, Di Bello and Kindred collaborated with NYCT on the design of a three-level evaluation to measure change and to identify trouble spots while the MIDAS implementation was in progress:

1. Cognitive probes,
2. Analysis of usage patterns using log-in records and keystroke recordings, and
3. Analysis of financial impact as measured by mean distance between failure (MDBF is the number of in-service vehicle failures divided by the in-service distance traveled).

A detailed discussion of each of these methods is beyond the scope of this paper, but a brief description of these assessments will render the results more comprehensible. First, the trainees were tested on their understanding of scheduled maintenance concepts using a specially designed cognitive battery. The battery was a set of tasks constructed to get at implicit, spontaneous strategies, and not simply memorized verbal material. This method of getting at implicit knowledge in a target domain was developed in previous research on DSSs and proven highly sensitive for getting at in-depth knowledge gained from experience. It was adapted to this project by redesigning the battery to target reactive and proactive ways of analyzing machine breakdown (*3*). The testing was conducted in one-on-one interviews with volunteer mechanics. Second, usage logs were downloaded from MIDAS and analyzed for complexity and accuracy of data input. For this analysis, data input by mechanics who participated in CUNY workshops were compared with data from a control group (mechanics who were trained by vendors or on the job, but who had equal work-time experience using MIDAS). The first of these logs was collected 7 months after MIDAS went live and continuously thereafter. Third, the depot's MDBF for the months before and after MIDAS implementation was compared with that of other depots where MIDAS was not yet implemented.

The pilot project was judged largely successful. Follow-up, one-on-one testing was done 6 months after training. Mechanics who attended the workshops had a much deeper understanding of MIDAS and scheduled maintenance concepts and were transferring this knowledge to their thinking about the buses they worked. In striking contrast, the control group continued to conceptualize defects in repairs within a traditionally reactive paradigm (Table 2).

Downloads of workers' navigation through the system and data entry practices also showed that trainees entered component data that were more than twice as detailed as the control group. There was also considerable within-group consistency in the type of detail entered: That is, statistical analysis showed little within-group variation even

TABLE 2 Cognitive Battery Results After 6 Months

	Real Bus		Toy Bus		Machine T	
	Trad.[a]	S.M.[b]	Trad.[c]	S.M.[d]	Trad.	S.M.
Control Group	.58	.39	.63	.38	.43	.32
One-Day Work Shop	.42	.58	.43	.51	.34	.48
Two-Day Work Shop	.38	.62	.37	.65	.35	.60

[a] ANOVA indicated significant differences between controls and 2 Day Trainees; F(2,27) = 5.70 p< .008

[b] ANOVA indicated significant differences between controls and 2 Day Trainees; F(2,27) = 4.32 p< .02

[c] ANOVA indicated significant differences between controls and 2 Day Trainees; F(2,27) = 4.48 p< .02

[d] ANOVA indicated significant differences between controls and 2 Day Trainees; F(2,27) = 6.68 p< .005

Cognitive Battery Results after six months; Proportional Scores on the "History" tasks, scores shown by sub-task. Note: both training groups were identical to the control group before training. Post test scores show a marked decrease in "traditional" reactive maintenance strategies and a marked increase in proactive "Scheduled maintenance" strategies after training. The effect was strongest with those who underwent two days of training. This effect transferred across objects.

"Trad." indicates proportion of all possible traditional strategies performed; "SM" indicates proportion of all possible proactive preventive planning strategies performed.

though the input was much more complex. In contrast, the control group entered much less detailed component information and showed for greater idiosyncracy in their choices of what to enter (Table 3).

Examination of the detail level of codes entered showed that the trainee group consistently used more detail-specific codes when identifying the components targeted for repairs (Table 4).

Last, MDBF for the pilot depot was higher overall but, more importantly, showed a gradual incline, whereas non-MIDAS depots, and NYCT as a whole, were experiencing a decline (see Figures 1 and 2 for comparisons). Qualitative data supported the general trend. For example, more than 200 design changes in the system were worker-initiated after the pilot. In addition, workers' complaints about the system and its deployment show a sophisticated understanding; they wanted more terminals close to their work benches and asked that the data tables reflect more detail so that they could better describe their work. Vandalism of terminals left in open work areas—a concern of upper management—has not occurred at all.

CONCLUSION

There have been many reasons invoked for the frequent failure of technology implementations, and most acknowledge that the "user" is a significant factor. In this paper we have sought to show that involving the user can actually increase the chances of success and give some stability and depth to the technology's overall impact. However, as should also be clear, "considering the user" is not a simple matter of a few presentations, surveys, or even traditional training courses. Instead, this paper proposes an iterative implementation process that incorporates alternative models of worker education, project evaluation, and ongoing application modification on-site. These components are considered to be key issues and yet they are not easy to address. On the other hand, the payoff is well worth the extra work. For example, it is estimated that the educational and design modification processes composed, at most, 10 percent of the total project budget (this includes the cost of overtime used to replace workers being trained) and yet drove its chances for success from 40 to 90 percent. Since NYCT is already seeing a financial payoff (Figures 1 and 2), this seems a small price to pay.

ACKNOWLEDGMENTS

Special thanks go to Ed Tella and Dave Slater of TENERA, Inc.; Dave Merlin of IBM; Project Director Douglass Roby, Chief Maintenance Officer John Walsh, and Senior Vice President Butch Seay of NYCT Department of Buses; and all the members of the MIDAS project (Ted Bonczek, Vinny Raco, and Tony Mastellone), without whose help this paper would not have been possible.

TABLE 3 Average Frequencies of Component Codes, April 1995

	mean	standard deviation
Control group[a]	1.70	1.07
1-Day Training	1.55	.52
2-Day Training	1.34	.27

[a] Control group was vendor trained and had been using the system on the job as long as the workshop trained groups.
System went "live" October 1994.

Average frequency of using a particular component code in response to bus-driver determined symptom codes; lower mean indicates greater variance of response and more detailed and accurate data. Small standard deviation indicates a homogeneity effect.

TABLE 4 Users' Choice of Component Codes To Identify Problems

	Most general <—				—>Most specific	
Group	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6
Control Group [a]	10	24	48	57	16	9
Trainee Group	10	33	62	126	25	15

[a] Control group was vendor trained and had been using MIDAS as long as the workshop trained group.

Users' choice of component codes used to identify problems in vehicles six months after workshops and system implementation. Numbers indicate the number of different codes used by each group at each level of database detail. High numbers of codes in the higher levels indicate more detailed data entry.

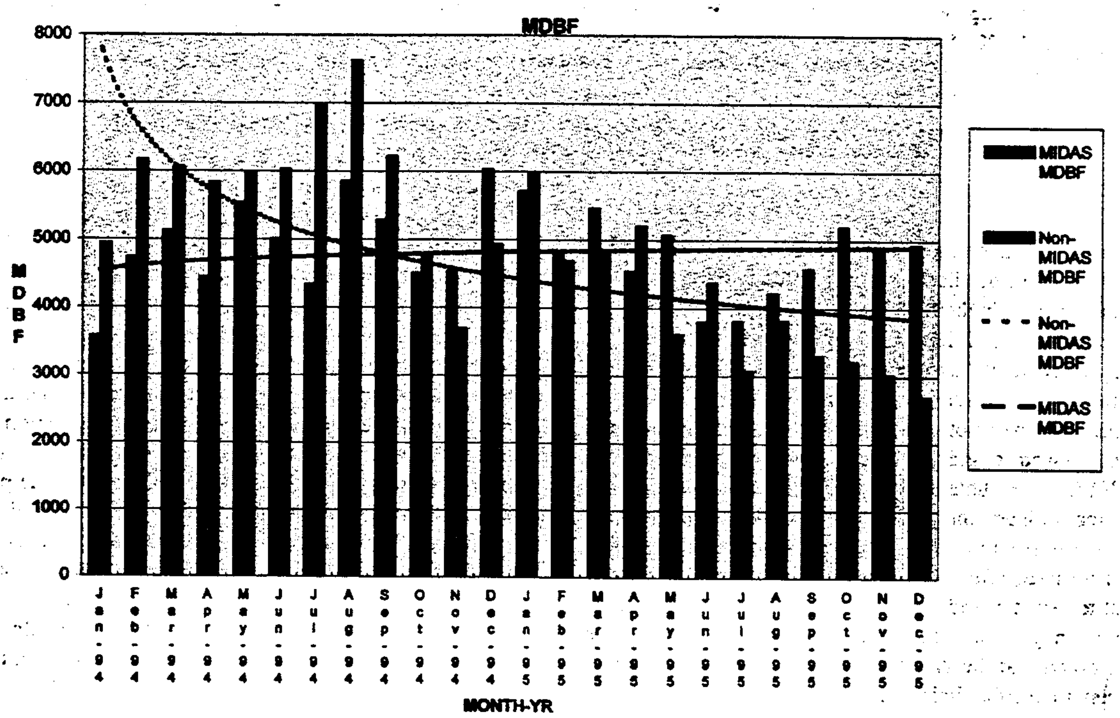

FIGURE 1 MIDAS versus non-MIDAS depot.

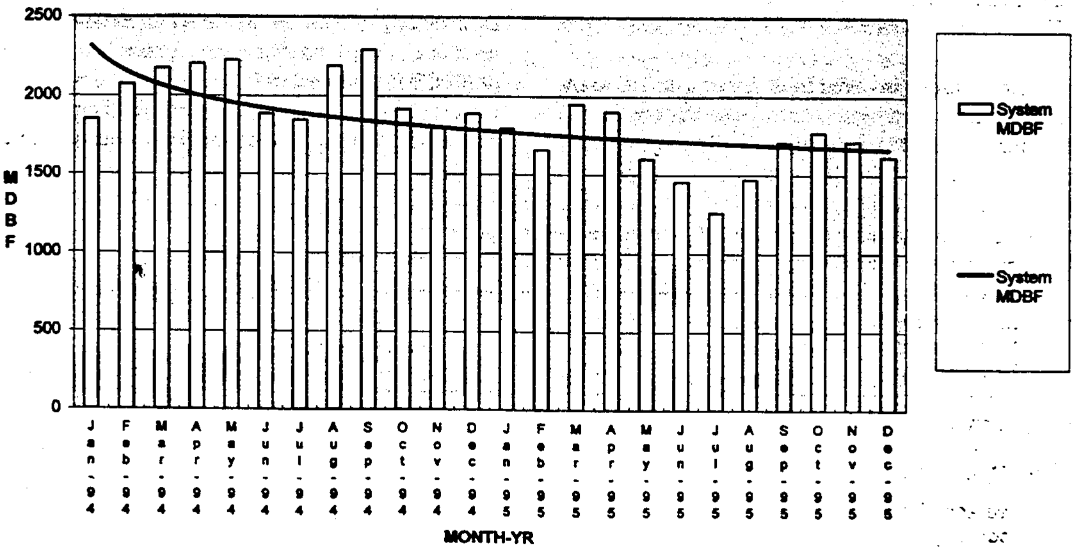

FIGURE 2 Surface transit MDBF (all depots).

REFERENCES

1. Di Bello, L., and J.-C. Spender. Constructive Learning: A New Approach to Deploying Technological Systems into the Workplace. *International Journal of Technology Management*, 1996.
2. Di Bello, L. Providing Multiple "Ways in" to Expertise for Learners with Different Backgrounds; When It Works and What It Suggests About Adult Cognitive Development. *Journal of Experimental and Theoretical Artificial Intelligence*, Special Issue, 1996.
3. Di Bello, L. Exploring the Relationship Between Activity and the Development of Expertise; Paradigm Shifts and Decision Defaults. In *Naturalistic Decision Making* (C. Zsambok and G. Klein, eds.), 1996.
4. Di Bello, L. Education for Complex Workplace Technology; Unpacking the Assumptions of Training That Fails. Presented at 2nd Workplace Technology Conference, Russell Sage Foundation, New York, 1996.
5. Dreyfus, H. Intuitive, Deliberative, and Calculative Models of Expert Performance. In *Naturalistic Decision Making* (C. Zsambok and G. Klein, eds.), Erlbaum, N.J., 1997.
6. Kindred, J., and L. Di Bello. Reframing In-Depth Learning and Development: Comparing Constructive Activities and Knowledge "Replacement" with Traditional Training and Knowledge "Acquisition." Presented at Annual Meeting of the A.E.R.A., San Francisco, Calif., April 1996.

Publication of this paper sponsored by Committee on Transit Fleet Maintenance.