SOFTWARE MAINTENANCE
Case Studies

David Bellin

Department of Computer Science, William Paterson College, Wayne NJ 07470

ABSTRACT
The specific needs of the smaller computer user for software maintenance are discussed. Implementation problems and third party support are considered. The major areas of concern are seen to be preventative maintenance, system backup and recovery, application enhancement, and use of third party expertise. Several actual case histories are discussed, and the cost of failure is addressed.

INTRODUCTION
Our developing field of software maintenance can today begin addressing the concerns of the user of micro computer systems. With the dramatic growth of computer systems in the $5,000 to $20,000 range, it is necessary to summarize what experience with minicomputers has generated.

During the last explosion of computer usage begun fifteen years ago with the minicomputer, salesmen often touted mini's as the answer to the programming headaches too often found by large computer users. Too often this was, of course, not the case, and poorly signed, developed, and ill-maintained systems led to many publicized horror stories and legal battles.

Are we about to repeat this same process with the micro boom? Or can we summarize some of the pertinent factors of software maintenance learned during the minicomputer growth period? If we can study case histories and develop an expertise or sensitivity to areas of concern in maintaining micro computer software, not only will the micro computer boom far surpass that of the mini's, it will also be much less painful for the computer user.

It is generally as an afterthought that the micro computer user becomes aware of the need for ongoing software maintenance. Micro and minicomputer installations often take place within a fixed budget, turnkey approach. Often the only ongoing expenses quoted by the vendor and budgeted by the purchaser are allocated to supplies and hardware maintenance. The computer system, hardware and software, is sold on the premise that once installation is completed, the user merely has to 'turn the key' in the morning and the machine will practically run itself.

This frequently is not the case. And it is a very costly discovery for the user that the software must be backed up, enhanced, and carefully evaluated.

The cost of failure is not the cost of replacing the system, but is the cost of losing the business that depends on the computer system. This may be crippling for a small organization.

As it affects the smaller firm, we can see four major areas of concern: 1) Preventative maintenance, 2) System back-up and recovery, 3) Application enhancements, and 4) Use of third party expertise.

PREVENTIVE MAINTENANCE
Several important topics must be included in discussing the prevention of a software system breakdown. Operating system updates issued by the manufacturer must be evaluated and applied to the user system and so must revisions to the language translators or compilers and utility programs used at a computer site. One medium size cosmetics called the author when the programming house which developed their software went into bankruptcy proceedings. The operating system was a version no longer supported by the computer manufacturer, and no longer in use by other programmers. This alone made further maintenance next to impossible without installation of the latest operating system. In such a case, the possibility of severe disruption to the user exists, due to possible incompatibilities in system revisions skipped over the years. Had the original software house installed the updates at regular intervals, any such problems would not only have been resolved earlier, but also with far less disruption to the user.

Similar considerations hold true for language processors. Language systems not supported by the original equipment manufacturer are frequently installed. Regional software houses will develop their own unique language translators. Our cosmetics company mentioned above has their problems exacerbated when they suddenly realized this was the case at their firm. Not a single consultant they contacted when the original programmers went bankrupt had any experience with the proprietary version of BASIC developed by the installers. Without exception, the first course of action suggested was to convert every program to a standard version of BASIC. Needless to say, the cost...
this process was to be substantial. With foresight into the demands of software maintenance, this could have been avoided.

One case of many shows how important language processors are. Consider, for example, the Data General NOVA computer. In addition to DG RDOS, over three other operating systems are sold by third parties. Under RDOS alone, at least five different versions of BASIC are available, in addition to that supported by Data General. However, taken all together, they are used on a minority of Data General installations. Should the small firm need a third party review of the computer, how easy will it be to find a firm with the needed expertise? Or to take over the software development?

Preventive maintenance should also include a regular inspection of the software system by an outside expert. The smaller firm has limited expertise. Minor programming is usually done part-time by someone with other areas of responsibility. The structure and integrity of disk files should be inspected. File usage must be estimated and projected into the next few months. The consultant should verify the proper performance of back-up procedures.

This should be summarized in a written report each month submitted to management. Avoid the situation of one small manufacturer who was suddenly told by his programmer, "Next week the order file will fill all available disk space and we will not be able to enter new orders." This resulted in the hasty purchase of an additional disk drive, at extra expense. It also had the effect of forestalling a needed review of the entire installation, and delayed completion of a forecast of hardware and software needs for the coming year.

The last area which should be included as part of preventive maintenance is ongoing training. A few sessions for the computer operator when the system is installed is not enough. Data entry clerks come and go. Operations manuals find their way into the corner to collect dust, while such training as exists is done by word of mouth as the new clerk enters his or her first orders. If this practice continues, a degradation of operations knowledge and practices will occur. Two years later, the sixth new operator thinks he is doing a backup each night, ignoring the cryptic error message "COPY aborted block 53139" because "it never seemed to make any difference." Management is in for a lot of fun when the disk heads crash and they discover none of the backups are good. If you think this doesn't happen, work as a mini-computer consultant for a few years!

This situation could easily have been prevented with regularly held one or two hour classes for operations personnel. Once a script is developed, it can be refined as system requirements change and used over and over again. Preventive maintenance, if considered as above, can do a lot to ensure the smooth operation and planned expansion of the smaller computer system.

SYSTEM BACK-UP AND RECOVERY

An auto retailer was using his minicomputer to track inventory, process orders, and write invoices. Four stores using telecommunications were dependent on the central minicomputer. The system had worked out so well that the stores relied on it—and were screaming whenever it was shut down for routine maintenance. The operators backed up the data files weekly. When a sealed disc was added to the system they were even more careful to copy to a removable disc each day. Without warning, one hot summer day, the machine stopped processing. None of the operators could restart the system.

The manufacturer's serviceman arrived on the scene, analyzing the problem as a faulty controller board which had destroyed the disc data. All the user had to do was reformat the pack, restore a backup of the system, and run. Only it turned out that the backups only contained data files, but none of the programs themselves. No backup copy of the programs was to be found. Nor did the programming house have any—all they had were some hardcopy printouts.

It may sound impossible, but it actually happened to a firm with several retail outlets. They were lucky—a software whiz was able to retrieve most of the programs, after three days at $75 per hour. They could have gone bankrupt, if unlucky.

What could have been done to avoid this situation? The best backup procedure on paper is meaningless unless it has actually been tested by attempting to recover a lost system. The discs can intentionally be wiped clean. Then verify that the backups which have been made contain all the necessary information for restoring a fully functional system. And be sure that the computer operators on-site know not only the backing-up procedures, but also the restoration process. There should be no need to call an outside expert to restore a system to operation after a hardware malfunction is repaired.

I should also note that assurance of a proper audit trail of all work done, on paper, falls within the software maintenance framework. There will always be some data to be re-entered since the last backup was made in a small system. There must be a proper log of work done so the correct source data may be retrieved and re-entered.

APPLICATION ENHANCEMENTS

It is typical for any data processing application to evolve with time. Laws change, a new product line is added, additional reports are needed, 'bugs' are found in programs run without apparent problems for years. The evolution may be so dramatic as to warrant complete replacement of whole modules by new programs.

The smaller user is generally unprepared to accept this view. The "turn-key" concept goes against continuing expense and change, and it was a point emphasized at time of purchase in many cases
I find, therefore, that software enhancements are left until the last possible time in many cases. 'Bugs' will not be resolved until operations are brought to a standstill, even if they are already known to the programmer. Management will be reluctant to accept the expense of a total re-write of patch-and-quit programs and will opt for the quick fix as the least expensive way out in the short run.

Where significant enhancements are added to the software package such as new management reports or a word processing module, costs can be estimated in advance by software personnel. Expense can be justified in terms of the company's information needs. It is important that the software maintenance staff be sure to write down a detailed specification of the new modules for approval, including sketched report formats. This will prevent endless rounds of changes and re-programming. I have participated in such rounds which have lasted up to 18 months.

Re-documentation of any changes to the software system is an absolutely essential and often overlooked part of maintenance. Joan was the chief data entry clerk for two years at one firm. She was there when the new field label "EXP" was added to the order entry screen and knew the only valid entries were 0 or 1 (even though any number was accepted), and that anything else would be ignored. When she retired and moved to Florida, nobody could figure out what to put in that field and two weeks of chasing were wasted because the programmer did not maintain the operations manual along with the programs.

USE OF THIRD PARTY EXPERTISE

The small computer owner has no choice but to rely on an outside party for software maintenance. The expense of $40,000 for an experienced systems analyst or $25,000 for a competent programmer with expertise in the particular system cannot be justified if the total operating budget is on the order of $10,000-$20,000 per annum.

In order to obtain the expertise needed for the maintenance and enhancement of the computer system, it is advantageous to formalize the third-party relationship with a long term contract. The software contract is in many ways analogous to that offered for hardware maintenance. It should include regular preventative maintenance (as discussed previously), and minor program correction and enhancement up to an agreed upon number of hours per month. Provision for rapid phone response in emergencies is a must. Often the maintenance vendor will elect to install modems, the cost either being absorbed by the vendor or shared with the user.

An on-going maintenance contract is of benefit to both sides. For the user, it guarantees written reports on the condition of the software system and response when program errors halt processing. For the software vendor, it guarantees a certain amount of income and a number of productive hours for the coming months. Both sides must be confident that the software maintainers have depth and experience in the particular hardware/software combination being used. It is the author's feeling that while these agreements are not yet common practice for the micro or mini computer installation, they will become increasingly prevalent over the next few years.

SUMMARY

It is apparent from case histories of the mini computer upsurge that the firm purchasing a micro computer can no longer afford to ignore the costs and responsibilities of software maintenance. Hard cash must be allocated to the repetitive needs of backup and program maintenance.

What is the cost of falling to do this? Perhaps that is the best way to pose the question. For if your business depends on your computer, it doesn't matter how small the computer is or what the dollar figures in the receivables are. You will still not be able to get at them if you have no way to restore your system from a backup. Is it really "less costly" for a 10 employee firm to go bankrupt? Certainly not to those ten people!

Others have commented on this topic:

One reason for the programmer shortage and related rising costs is the compelling need for ongoing maintenance once a package has been put on the market or developed by the users themselves. Maintenance includes error correction, requests for modification of software, and instruction on how to employ the software. Only 5% of assistance calls concern error correction; the vast majority of calls are requests for modifications and instructions on utilization.

Therefore, I would formulate a policy for justifying software maintenance procedures in terms of the potential cost of a failure. This contrasts with the traditional approach evaluating maintenance costs solely with respect to the initial purchase price of the computer hardware.

I close by pointing out that properly performed, software maintenance will ensure continued productive use of the small computer system. The frequent disasters and problems we have seen in the years of minicomputer growth can be learned from and prevented. Both end users and software technicians are coming to accept that data processing systems are defined by a constant state of change. Information needs change, volumes of data change, and entry procedures change. The job of the software maintenance expert is to predict the areas of change and accomodate them creatively.

REFERENCES


ACKNOWLEDGMENT

The author would like to thank System-Aid Computer Control, Inc., PO Box 925, Flushing NY 11354 for their practical assistance in the cases mentioned, and Mr. N. Zvegczintov for editorial assistance.