MANAGING THE SOFTWARE/PEOPLE TRADE-OFF

Thomas D. Lutz, Mayo Foundation

The objective of this paper is to give scientifically and technically oriented individuals some insight into the difficult management process of choosing between software packages and people. Although the management community wants to make decisions based on the scientific analysis of facts, maximizing resource productivity is often the result of perplexing experiences and intuitive judgments. Consequently, this brief document will expose the reader to such non-scientific parameters as emotion, environment, profit, expenses, productivity, pride of authorship, verbal communication, etc.

As Dr. Gerald H. Weinberg so aptly states, "Computer programming is a human activity... Either you can program or you cannot. Some have it; some don't." As a result, an investigation of the software/people trade-off is really a question about people - and using the people resource most effectively. It should be realized that the term "most effectively" cannot usually be measured in terms of productivity or the best program for the least dollar. For example, if the most efficient program is developed by a technique which leads to the emotional destruction of its builders or users, then we could hardly say that it was developed 'most effectively'. Hence, the trade-off question is a management problem.

Before the actual trade-off is analyzed, it might be appropriate to define several of the terms.

Software - A generalized collection of programs which are designed to enhance and usually simplify the process whereby people "persuade" a computer to do their bidding.

People - In this paper, people are defined to be those individuals who are directly involved with the problem of coercing a computer into following their every request and responding to their every request.

Trade-Off - The artful task performed by management when confronted with the problem of producing a maximum amount of meaningful results within limited resource constraints without destroying the ability of those resources to continue functioning when future demands are made.

System - "... a composite that functions as an entity, and its main attribute is that its utility, or power, is greater than that of its components summed individually."

The components of a simple computer system include at least; hardware, software packages, application programs, programming staff (people), data, and users. The objective of the management process in the software/people trade-off is to manage and support the people resources in such a way as to minimize the cost while satisfying user demand and keeping the programming staff enthused and working. When this occurs, the system is indeed more productive than the productivity expressed by the sum of each of the components.

The Use of Software Systems

By definition (stated above), a software system should assist people with directing a computer to respond to their information needs. Three types of software systems will be considered here for purposes of illustration:

1. Compilers - Those collections of computer programs which translate people-like languages into the 'mystical languages of computers'.

2. Data Management Systems - Programs which assist people in storing data in some pre-determined structure so that people with poor memories can recall forgotten data which can be arranged according to new and different structures.

3. Application packages - A collection of programs which assist people in analyzing previously stored information according to a pre-determined methodology quite unique to a given application area.

The Statistical Analysis System (SAS) most clearly applies to category type 3. That series of programs is used to assist people involved in statistical analysis by providing pre-developed programs (software) to relieve people of considerable programming effort. Programmers/Analysts/Statisticians are freed from the shackles of detailed programming so that they can address the data integrity and scientific value of an analysis of variance or multiple regression analysis problem, for example. It is worth noting that in scientific and statistical research, the scope, approach, and parameters of a problem often change with the data so that statistically meaningful results coming from the computer often yield the by-product of a discarded program. Further analysis usually results in the development of new programs. In the computer industry, this is often referred to as 'one-shot programming' and a statistical package allows for the increased efficiency of people by shortening the time to get meaningful results from the computer. New programs yield new programs which in turn yield new problems. The iterative process of data - programmer - results - data - programmer... is a critical and timely cycle with much of the emphasis being placed on the word timely.

Let's review an example from software category type 1. Several years ago, a highly skilled programmer was asked to develop a system whereby a secretary could perform word processing functions at a typewriter-like terminal. The intent was to create a program which would allow the secretary to type documents with the ability to automatically justify right margins, rearrange words, sentences, and paragraphs, delete, correct, or add words, phrases, and sentences and produce a finalized copy with a minimum amount of retyping. The programmer, having considerable skill and having logged thousands of hours as an APL (that's one of
those translators referenced in type 1.) expert, was able to produce a word processing system in the matter of a few days. Facing a critical deadline to produce a complex system resulted in reducing people's time and increasing productivity by the application of a software package—did it truly increase productivity? Hence, it is important to examine carefully a 'contra-area' of management concern.

The Abuse of Software Systems

If we review the definition of a system, we recognize that several components work together to improve total productivity of their collective efforts. One of the components described in the system above is the programmer. At least three other components must be considered; computer hardware, secretarial time, and money. In general, a system such as word processing might be in use eight hours each day by a secretary. The computer and other components must be considered; computer management concern.

The Abuse of Software Systems

If we review the definition of a system, we recognize that several components work together to improve total productivity of their collective efforts. One of the components described in the system above is the programmer. At least three other components must be considered; computer hardware, secretarial time, and money. In general, a system such as word processing might be in use eight hours each day by a secretary. The computer would be called upon to perform its functions again and again and again...

Often, the use of software systems improves the productivity of programmers (providing they are well disciplined) but generally degrades the performance of the computer system. Simply stated, 'something's lost in the translation'. The translation prepared by the APL3 compiler is not nearly so efficient as the results of experienced programmers who write directly in the languages of computers themselves. The word processing systems developed through the use of APL required considerably more computer hardware, ran much slower, and did not take advantage of some of the computer features which could have reduced some of the work involved.

If the problem to be run, in this case the typing of documents, were done infrequently (let's say once per day), then programmer speed and efficiency is obviously more important. However, management is faced with a serious collection of decisions when a particular program may be run 500 times per day and the costs of hardware begins to outdistance the cost of a programmer.

Even a package such as SAS can be abused. In addition to being a valuable tool in the statistical analysis of data, it can be used as a language and a database management system. Ingenius SAS programmers can often perform Herculean tasks for which SAS was not intended. (It just may be that the SAS Users Group request for "darn cute code" may produce at least as many abuses and inefficient applications of the package as it produces meaningful and valuable routines.)

People Versus the Computer

Management is then faced with a significant dilemma. The software, when applied properly, can increase the efficiency of a "total system" while abuse of the package can decrease total productivity. The trade-off decision would be far simpler if programmers could assist in the evaluation of a software package in an objective and impartial way. A problem often arises however when programmers become attached to particular software, become so expert in its use that they will use no other, and consequently apply the package in all cases—both good and bad. Users sometimes contribute to the difficulty of making a decision because their needs are vaguely defined and applications which originate as 'one-shot jobs' ultimately end up being run 500+ times per day.

Adding to the dilemma is the fact that the cost of resources continually fluctuates. Computer and software inefficiencies of ten years ago resulted in significant cost to an organization. Today, however, hardware efficiencies are improved by several orders of magnitude. In the meantime, the cost of programming people has risen and users are taking matters (and programmers) into their own hands. Highly paid scientists and executives have become programmers through the use of software packages (which often compounds the system inefficiencies).

A technique employed to assist in an analysis of the trade-off problem is to seek some common parameter which can be used to evaluate both the components and the system itself. In many cases, organizations may wish to relate everything to financial cost. The problem is that not all costs or benefits can be articulated in terms of dollars. For example, assume that a particular problem is analyzed carefully and the cost required using a software system is estimated to be three man-months (which can obviously be translated into dollars) while the same application can be written in computer language with an investment of five man-years. At the same time, the computer efficiency of writing in computer language could be a ratio of 20 to 1 over utilization of the software package. With these facts, organizations attempt to apply people cost and hardware cost to determine the proper approach based on the volume of computations. But other factors must be considered before the conscientious manager can make a responsible decision.

1. Are enough people available to expend five man-years before the dynamics of the problem change? In other words, can the job be programmed soon enough to provide useful results for the corporation?

2. If the computer language route is taken, will the program and its associated documentation lend itself to change as the needs of the users change? (Sometimes, applications are relatively static so changes are not a problem.)

3. Does a software system provide added flexibility so that the program can be processed on a variety of computers? By using the package, is the organization locked into a particular computer? (By writing in the computer's language, this flexibility is usually lost.)

4. Is expertise available in both approaches so that a management choice really exists?

5. Is the present or planned computer system large enough and fast enough to give timely results if the software package is used?

6. What is the mental, emotional, and training status of the programmers? (Their adaptability, ability, and interest must be considered.)
Organizations and management have a responsibility to evaluate the intangible, nonfinancial human characteristics in the use of computers and the application of software systems. Norbert Wiener provided a fitting quote to summarize the dilemma.

"A few leading computer scientists have been alarmed by humanistic neglect in the development of computerized systems, and they have warned us that human responsibilities will grow more difficult and more complex, that the enormity of the challenge for the 'human use of human beings' could eventually overwhelm humanity."

Hopefully, the reader is by now beginning to sense that analytical analysis is often marginally helpful in making decisions regarding the software/people trade-off.

Management's Role Is Balanced Judgment

Management is often caught between the proverbial "rock and hard place". On the one hand, users want their newly defined computer work completed 'yesterday' but do not often become involved in the resulting impact upon systems people and hardware. At the same time, the people responsible for the operations of production systems resist new jobs which may upset the balance of a finely tuned system.

An approach used by systems management in an effort to 'get out of the middle' is to involve users in the dilemma. This is often accomplished by stressing the point that there are no "free lunches". Development and operational costs are passed back to the user. In general, however, the trade-off decision is still a matter of management's perspective of the parameters involved.

Hopefully, this brief treatment of the software/people trade-off has given scientifically and technically oriented individuals a greater appreciation for the balanced judgment which is required of management as complex choices are made.

However, the issue is complicated further since hardware and software vendors are interested in expanding the use of software systems.

1. Software systems generally provide organizations and people with the ability to more quickly and completely produce computer solutions to information problems.

2. More rapid development of computer programs leads to a faster expanding load upon the hardware system which generally leads to an enlargement of the equipment. In many cases, the utilization of software support does not maximize the efficiency of the computer as much as it attempts to maximize the efficiency of people. Hence, hardware utilization is again increased.

Information science professionals are aware that hardware manufacturers are continually improving upon price-performance. Yet increased numbers of computer applications and the employment of general purpose software systems have created a computer industry growth curve which continues to startle the stock market. (One well known computer manufacturer climbed from 9th to 7th in total revenue on FORTUNES top 500 list between 1974 and 1975.)

The systems specialist is usually interested in producing more and improved results for the organization he/she serves but that specialist can become addicted to change, enamored with software packages, and instilled with an attitude that no problem is too difficult to solve with a computer. Corporate executives are presenting a "hard line" approach to computer decisions and insist upon a meaningful evaluation of the alternatives. As the manager sits between the system specialist he directs and the corporate executives he serves, one of the most helpful tools in making these difficult judgments is a team of scientifically and technically oriented people who understand the dilemma and participate in the solution with a non-parochial perspective. Pride of authorship, emotion, programming fever, and a narrow perspective must be shelved to make sound, defendable decisions regarding the software/people trade-off.

References


