Title
ENCOURAGING MATURE USER BEHAVIOUR IN THE END-USER COMPUTING ENVIRONMENT

Author
Stevens, D.F.

Publication Date
1988-06-01

LBL-25279

Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA, BERKELEY

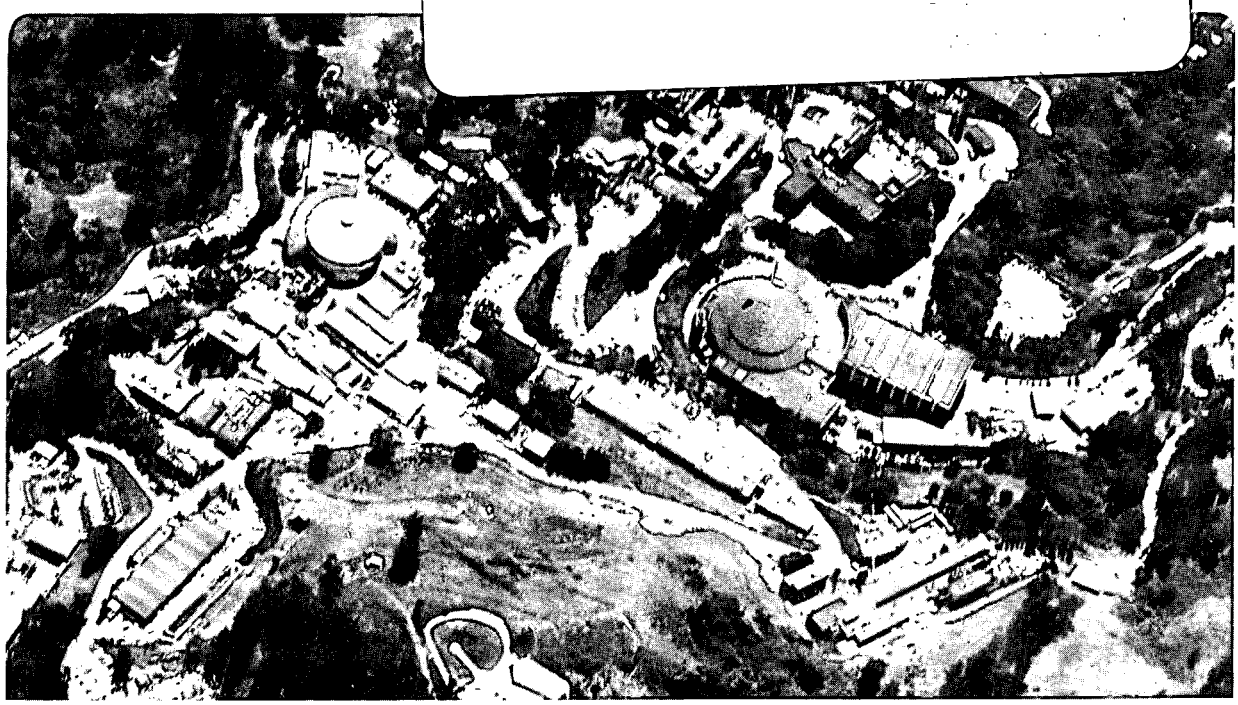
Information and Computing Sciences Division

To be presented at the 1988 Pergamon Infotech State-of-the-Art Review, London, United Kingdom, November 16–18, 1988

Encouraging Mature User Behaviour in the End-User Computing Environment

D.F. Stevens

June 1988



Prepared for the U.S. Department of Energy under Contract Number DE-AC03-76SF00098.

DISCLAIMER

LBL-25279

Encouraging mature user behaviour in the end-user computing environment*

David F. Stevens
Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

June, 1988

Prepared for the Pergamon Infotech 1988 State-of-the-Art Review
London, 16-18 November, 1988

* This work was supported by the U.S. Department of Energy under Contract DE-AC03-76SF00098

Encouraging mature user behaviour in the end-user computing environment[1,2]

David F. Stevens
Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

Introduction

Truly revolutionary technology appears to require a considerable time to diffuse widely enough into the "old" society to prepare the way for, and develop the infrastructure necessary to support, the changes that will eventually justify the term *revolutionary*. Our experience with the telephone and the automobile indicates that, in the modern era, this time is about two generations, or roughly 40 years. Revolutionary technology also exhibits a characteristic common to all powerful tools, revolutionary or not: their potential for harm is as great as their potential for good. (You can cut off your arm with a power saw more easily and quickly than you can cut through a 12" beam.) Computing is generally considered to be a revolutionary technology that has recently passed its fortieth birthday. "End-user computing" may be among the first of the revolutionary changes that we can expect eventually to see, or it may merely be evidence of the diffusion of the technology; in either case it provides a broad segment of the work force with access to very powerful tools. It is the responsibility of all of us in the information technology (IT) professions to see that these tools are used wisely.

Mature behaviour

Mature behaviour is characterised by appropriate, effective, and efficient use of the available personnel and tools. In the present context, one is tempted to read "computers" for "tools", but that is not intended. The entire computing-oriented or computer-assisted *system* must be considered, including administrative procedures, corporate policies, and people, as well as computing hardware and software. The mature user uses all elements of the system appropriately and effectively, and strives for efficiency of the system as a whole. The literature is replete with exhortations on efficient and effective use of the hardware and, to a lesser extent, software. *Appropriate* use, on the other hand, appears to be assumed or ignored, as do the non-computer parts of the total system.

Appropriate use is a fundamental aspect of mature behaviour. A mature user uses computers to complement people, to enable them to be more productive; he does not use computers either as sophisticated toys or as sources of personal power; he does not waste resources or abuse special privileges or loopholes; he observes institutional guidelines and standards, particularly with respect to the protection and preservation of data; and he exhibits courteous behaviour towards his fellows.

This definition of mature behaviour is generic; it therefore applies to IT as well as to the users. The burden of this article is that IT has an obligation to accelerate the maturation process of the end users. That task cannot be accomplished in the absence of a continuing demonstration of mature behaviour by the IT staff. A full discourse on this topic is beyond the scope of the present article, but there is room for one basic principle: IT must live by the same rules that it imposes on the users. (If *you* can't get your work done under those rules, then neither can *they*.)

---

[1] This work was supported by the U. S. Department of Energy under contract No. DE-AC03-76SF00098

[2] This article draws upon or extends ideas presented in the following earlier papers by the same author: *Encouraging mature user behavior; Supporting the literate user; A look at white-collar personal computing;* and *On the rôle and function of a User Support Group.*

The importance of mature behaviour

If end-user computing were destined to be short-lived, of narrow scope, or of small impact, there would be scant need to encourage mature end-user behaviour. It seems clear, however, that the reverse is true: There will be significant impact; the scope will be broad; and the phenomenon will far outlast the tools and systems now in use.

The impact of end-user computing will be felt sociologically and organisationally, as well as technologically. It is likely that there will be significant surprises in all three areas, so the best we can do as IT professionals is to be ready for change. By *sociological* impact I mean the changes in office relationships that will accompany the introduction of new technology into each corporate environment; by *organisational* impact I mean the structural changes that will follow the dissemination of new skills throughout the organisation; and by *technological* impact I mean the changes to the nature of work (as distinct from acceleration or redistribution of existing work) in the corporation. There is not space here to provide more than the briefest glimpse at each.

Much of the real power in any organisation is wielded by those who control access to people and information. As end-user computing spreads through an organisation, new ways of access will develop. In many cases, some or all of these ways will bypass the current guardians; as they lose control, some of their power and mystique will pass to those able to adapt to the new environment. As power shifts, relationships will change, often in unpredictable ways.

Organisational impacts will come somewhat more slowly, but as certainly. Paralleling the way in which the introduction of end-user computing is distributing the programmer/analyst function, the newer applications of end-user computing will distribute other organisational functions. Many publishing and accounting functions, for example, are already moving from their central havens onto desktops all over the corporation. The central publishing department at my home institution now prepares only about one-quarter of the published papers (they used to prepare them all); for the balance, the entire process, including the preparation of camera-ready copy, is handled on desktop systems by the end-users. I no longer have to appeal to the Purchasing Department for a lease/purchase analysis, for I can do one in a few seconds at my workstation. As these and other functions become widely distributed, organisational change is sure to follow.

We have already noted that computing is considered to be a revolutionary technology. The revolution in the making is a three-step process. The first step is mere acceleration of work: The same work is done in the same way, but faster. The second step introduces the first dimension of change: It is still the same work, but done in dramatically new ways. Desktop publishing is an example of this; we are still preparing the same sorts of documents as before, but most of the work is now done in the originator's office instead of in a central publishing office. The third step is the introduction of wholly new work: As work is done differently, new kinds of work will come to be done. A possible example of this in the publishing area is the introduction of *hypertextual* material, in which a document no longer consists of a serially accessible set of sentences, but a multiply-linked mass of material in several media.

There are many different kinds of end-user computing. Some of them, such as programming in Fortran by engineers, for example, have been around for many years. Most, however, are just coming into their own. We now see departmental computers of significant power and capacity being programmed and operated by end users; powerful computer-aided design stations; business, clerical, and scientific desktop workstations with more power than early mainframes, made seductive by the wealth of applications available; application generators and AI shells that produce operational systems; query

systems for use by high-level corporate executives. End users are writing programs, developing applications, and designing data bases. As a result, a large segment of the total corporate resource is at risk. It is estimated that more than half the computing power in the average US corporation now resides outside the central IT function in the hands of end users. While this is a significant investment, it is miniscule compared to the investment in staff time spent at computer-assisted workstations. It will not be long before every professional, administrative, and clerical staff member will have, or have access to, and use, a workstation for applications not controlled by central IT. The potential for loss due to *reduced* productivity, redundant effort, misplaced files, inefficient or erroneous codes and spreadsheets, and improperly protected data, to name only a few areas, is enormous. (It is illuminating to estimate how much time is spent, institution-wide, just waiting for desktop systems to respond, even at rather high levels of performance. Consider document-processing as an example. The following scenario is thought to be rather conservative:

4 page-preview or formatting passes per hour, at 5 seconds each:	20 seconds
6 SAVE operations, at 1 second each:	6 seconds
2 back-up operation, at 3 seconds each:	6 seconds
Print-oriented delays (print processing, visiting the printer, reloading it, etc.):	180 seconds
General system-response delays (half-a-second once each minute):	30 seconds

Even with zero delay for text entry, the cumulative loss of time is 242 seconds per hour of workstation time, or more than 6.5%. One can expect compile and processing delays for 4GL applications or the processing of *ad hoc* database queries to be significantly larger.)

The rôle of the central IT function

The central IT staff can take any of several attitudes towards the spread of end-user computing. The most obvious are enabler, observer, or opposer. As observers, IT simply watch what the end users are doing, offering neither support nor opposition. (We in IT "paid our dues" and learned the lessons of adequate backup the hard way; why shouldn't they? If they come to us, we will answer their questions, but there is no need to volunteer any assistance.) The largest potential drawback with such an aloof attitude is that the users are likely to find solutions to their problems that do not require the presence of a central IT facility or staff.

As opposers, IT fight the onslaught on several fronts, using all of the following tactics: They use the inevitable loss of economies of scale resulting from the distribution of computing to justify opposing the installation of resources outside of IT; they tell horror stories about mismanaged data (to justify keeping control of all data within IT); they put impossible restrictions on central access for end-user systems (in the hope that the difficulty of central access will give end users second thoughts about doing their own computing); they provide unwilling and inadequate support. Much of their propaganda is true, but it is likely to have as much effect as Canute beating at the waves to stem the tide; opposers are likely to end as victims, overwhelmed by the backlash that develops as the end users achieve independence.

As enablers, IT recognise the inevitability of end-user computing (after all, the cost of a personal computer is within every professional's reach), and expend every effort to see that it is done right, and that the corporation as a whole benefits from it. It is my belief that the central IT function is a service function, and that its primary responsibility is to see that the power of computing is constructively applied to the affairs of the corporation. This responsibility is best exercised in today's world

by helping end users get the most out of their investment in computing, wherever and however they decide to do it.

It is also in the enlightened self-interest of IT to adopt the rôle of enabler, for at least three reasons. One is simple political expediency; the end user in question may be a person of considerable corporate influence. The second is that, according to current experience, the more effectively the end users compute, the more the central computing load will grow; as you adjust your service offerings to meet the new demands of end-user computing, including convenient, high-bandwidth access to the central facility, you will find significantly increased demand. After all, demand is clearly a function of the number of programmers feeding applications into the system; by promoting end-user computing you are increasing the number of programmers many-fold.

The third reason to become an enabler is the education about new uses that your staff will gain in the training process. As we have noted, the introduction of revolutionary technology changes society. In the small, that means it changes the way your users do business, as they shift from merely speeding up the old processes to wholly new ways of working. These new ways will provide new demands for central IT services, if IT is perceived as a partner and is poised to offer support. One of the side effects of a well-administered enabling policy is that IT will be so poised and so perceived.

One of the most effective ways to become an enabler of this new technology is to become the end users' teachers and trainers.

Other sources of training

There are several possible sources of training for end users. Indeed, if one believes the advertising brochures, formal training is unnecessary. (To a certain extent, in some areas of end-user computing, this is true, but only for users who are already experienced in similar applications.) Possible teachers include the end user himself, colleagues, vendors, third party consultants or specialists, and the central IT staff.

The user, especially in those areas and applications where formal training seems unnecessary, is strongly driven by a desire to get on with the task. He *may* follow a prescribed course of instruction, but few people are sufficiently self-disciplined to carry such a course through to completion. Instead, he will learn enough to get started, and then learn by doing, by guessing, and by blundering. ("Experience keeps a dear school, yet fools will learn in no other.") This method has the advantage that the instructor is always available when needed, but the disadvantage that he knows no more than the student. It usually results in moderate-to-heavy use of the next two sources of instruction.

Corridor conversation with colleagues is the way many of us have received much of our technical education. This method is *ad hoc*, disjoint, excessively specific (the question of the moment is answered, but useful generalisations are rare), ephemeral (the answer is easily forgotten), and a burden on the rest of the staff.

Vendors provide two forms of instruction, (formal or informal) classes and (presumably helpful) documentation. Let us consider the latter sort first. It is rarely sufficient to rely upon system documentation, on-line or off, to provide adequate instruction. Unless the user already knows the jargon peculiar to the application *and* the vendor, it is difficult to find out what tools are actually available or what they do. Whereas in some instances the name of a tool or command may provide a good clue as to its function (*mail, readnews, kill*), others (*grep, biff, awk*) defy analysis by the uninitiated. (These ex-

amples are all drawn from a *unix*[1] environment, but similar difficulties seem to exist in all environments.) On-line assistance is strongly limited by the developers synonymy. To appeal to *unix* again, standard on-line documentation includes a kwic-like permuted index containing entries for all commands, and for all the non-trivial words in a one-line description of each command. This sounds adequate, but in practice it is not nearly so. For one example, the command to create a new directory is *mkdir*; it is listed under "make" but not under "create". For another, the usual commands to list or display the contents of a file are *cat* and *more*; neither is listed under either "list" or "display". If you don't already know what to look for, it is extremely difficult to discover anything useful.

For classes, we can consider vendors and third-party instructors together. The quality of the classes, of course, depends upon the competence of the individual instructors. Even with excellent instructors, however, these classes tend to be limited to imparting technical proficiency. As we shall see below, technical proficiency is but a small part of the subject matter that *should* constitute end-user training. It is because of this last point that I claim that IT, in its rôle of enabler of new computing technology, should become the trainers of the end users. In some sense, anyone can acquire the technical information and pedagogical skills to impart proficiency, but it is only the central IT staff who have the motivation to do a good job of imparting the other knowledge necessary to maturity.

Subject matter

Traditional computer-oriented training courses are concerned almost exclusively with skill transfer: the training is considered to be complete when the student can perform the application acceptably. If the goal is *maturity*, however, the training must be rather more comprehensive. In addition to skill transfer there are many quasi-technical matters, constraints, matters of judgement, and even ethical matters that should be included in the curriculum.

Quasi-technical matters

These are the many tricks of the trade that are relatively independent of, or peripheral to, the specific application being taught. Although they are part of IT lore, care must be taken to *not* teach them as if they are part of a computer science course. Remember that the users want and need to know how to do their jobs, but are generally not interested in unnecessary system internals. (Some of them, indeed, don't even want to learn *necessary* system internals. You should translate this material into non-technical terms, insofar as possible.) The material covered in this part of the curriculum ranges from extremely basic to rather sophisticated; not all of it is suitable for all students (see the discussion of different types of students, below). Generic topics in this part of the course are:

- how to start an application (which may need to include "how to turn a terminal on")
- how to get on-line help (including "how to tell what kind of on-line help is available")
- how to read the reference manual, if any
- which of several possible tools to use
- tutorials and templates for common applications
- limitations and pitfalls
- how to pipeline tools (i.e., how to use to output of one as input for another)
- how to save work, and how to revert to a saved copy (also, when and how often to save)
- how to exit from an application, and what is saved under each exit method
- what defaults exist and how to personalise them

---

1 *unix* is a registered trademark of the Bell Laboratories.

Ethical matters

This is perhaps the most surprising segment of the curriculum, but one that must be included if your end users are to develop maturity. Left to their own devices, end users will evolve their own standards of behaviour. Experience shows that electronic civilisation mirrors the wider world in that much grief and misunderstanding occur before these standards develop in such a manner as to improve the quality of life. Good electronic citizenship, like normal good citizenship, revolves about consideration for others. Because many modern systems give the impression that each user is alone on the system, this is a message that bears considerable repetition, in every conceivable context. In the E-mail context, for instance, a few standards are beginning to emerge, but have not yet achieved universal acceptance. Among them are: clearly identifying intemperate opinion (if you have insufficient self-control to eschew it altogether; some places use *flame-on* and *flame-off* as brackets for this purpose[1]); not proliferating copies and distribution lists; identifying yourself (not usually necessary in E-mail itself, for the system will do it for you, but good to remember for conferencing); never masquerading as someone else; and not using real-time (conversational) messaging (except in emergencies) unless all parties concur. In all contexts, it includes a proper respect for the copyright and licensing laws, and the exercise of due care when accessing corporate data or data belonging to any other user.

These kinds of ethical behaviour tend to be independent of any knowledge of the systems being used. There are also some whose explanations, if demanded, require discursions into system limitations, even if the "rules" themselves make no reference to them. These are the rules surrounding system overload. To use the E-mail case, there are two reasons not to automatically copy the whole world on all messages: You can overload the people and you can overload the system. Many other tools, if naively used, have a similar ability to overload the system; anything that sends a message, accesses a disk, locks a record, or makes a copy, can cause enormous difficulty if used unthinkingly. It is impossible to provide a complete list, but the principle is clear: If the system is not immune to accidental overload, then the novice needs some specific limits.

So far we have considered ethical behaviour on the part of the user, *qua* user. Some end users also need instruction in ethical behaviour for system providers, for they will become system managers for multi-user systems. This means teaching them some of IT's hard-earned lessons in change management, advance notice, truth in advertising, capacity management, special privilege, system development, etc. Most institutions have a wealth of such material. One of the ways in which IT can demonstrate its own maturity and, not completely incidentally, demonstrate its worth as a partner, is to codify this information and pass it along to those end users who have become system managers.

Matters of Judgement

One aspect of maturity is the ability to make wise judgements on the selection of tools and the division of work. There is a great tendency to overuse the interesting tools and to underuse the boring ones, to redo work unnecessarily (such as developing a new memo form to demonstrate your proficiency with the new document preparation system), to shift secretarial work onto professionals in the name of faster turnaround. (It is often gratifying to do it all oneself, but you should remember that you do it rather less efficiently than a good secretary, and at professional wages.) It is foolish to attempt to forbid such abuses, if, indeed, they are abuses, but you can try to explore the consequences with your students and work with them to establish truly productive work methods.

---

[1] Actually, content labels are a good idea in general, to help the reader to distinguish among fact, instruction, opinion, emotion, humour, etc.

On the more technical side, you will find people using application generators to generate six-line programs for the same reason. It is important to teach early on the value of wise tool selection; the best way to do this is to pick examples that reward the use of the tool selected, both in the sense that it simplifies the user's life and that it promotes effective use of total system resources (including the user and his staff).

Constraints

These are the limitations on total freedom that your users will encounter. They are actually special cases of the other kinds of subject matter, but sufficiently important to rate special mention. Some are absolute, because of the limitations of the configuration, technology, or the real world. (They will never get a tenth-of-a-second response time through a satellite link, for example.) Others are legal or ethical, such as the use of someone else's copy of shareware or proprietary software. Still others are conventional, organisational, or political. In dealing with these, it is important to be truthful in specifying the source; you lose credibility when you are detected in deliberate untruth. Maturity involves recognising the true nature of the real world and dealing with it effectively.

Populations of end users

Not all of your users are alike, and they are unalike in several different dimensions. These include organisational differences, such as level in the hierarchy or job classification; technical differences, such as equipment available, tools preferred, and applications attempted; and attitudinal differences. The training implications of the technical differences are obvious, and need not be discussed here. The same is generally true of organisational differences, with the addition that the higher up the corporate ladder you go, the less time the "student" will be willing to spare you for training; executive tools should be self-evident. Attitudinal differences are more interesting. You will encounter *technophobes* (who distrust this electronic wizardry), *technofreaks* (who look upon it all as the video arcade of life), and *appliers of technology* (who can take it or leave it, as appropriate). You must try to discern who is which, and tailor your training as much for their attitudes as for the tools and applications of importance.

Technophobes are often unwilling students, having been forced by circumstances into an environment with which they have little sympathy. They are not comfortable dealing with electronic images instead of physical documents; they are lost in the absence of "real" work product. They have no feeling for cause and effect, and are confused by the fact that simple actions (tossing out an extra copy of a document, for example) may be relatively difficult to perform (requiring multiple confirmations), while apparently complex operations (such as pagination with automatic update of table of contents and index) can be invoked quite simply.

For these people you must avoid all technologically-oriented jargon. No talk of bytes, bits, MIPSes, logon/logoff, etc. Talk to them in application-oriented terms. (If the inventor of the application hasn't created them for you, *you* create them.) Classroom instruction is worthless unless there is a terminal on each desk, and they can follow along step by step. They learn by doing and by imitating. Templates and cookbooks are very useful when training these people. It is important to show them how and what to save, and even more so to tell them what and when to discard. Their lack of trust makes them unwilling to throw anything away. They will in general welcome constraints in the form of procedural rules, and will willingly adopt ethical behaviour, but they will show little tendency to broaden their technological horizons.

There is an interesting variant of the technophobe who can best be characterised as a *Utopian*. Utopians are those unfortunate souls who believe the promises of salesmen and advertising brochures. They thus look forward to the introduction of computing as an exciting adventure, but with vastly unrealistic expectations. They are quite unprepared for the pitfalls and limitations scattered throughout even the best of applications, and are in danger of being very badly burned unless you can bring an element of caution into their approach. (Unfortunately, a remarkably large fraction of this population consists of high-level executives fresh from carefully choreographed demonstrations. It is extremely important to provide them with some gentle reality training as quickly as possible.) Utopians should receive the same instruction as the technophobes, with respect to both content and pedagogical techniques, except that a little less time needs to be spent on reassurance and more on cautionary advice. For example, where it was necessary to emphasise removal of clutter for the true technophobes, it is necessary to emphasise back-up and saving of work for Utopians, for every little failure catches them by surprise.

Technophobes and Utopians are transitional phenomena -- as we reach the stage when the whole work force has had early exposure to computer-driven tools and toys, they will vanish -- but they are with us still and should be assisted to work effectively to the best of our ability.

The technofreaks are those who are turned on by computing. It may be they, too, are a transitional phenomenon, but since we still have automobile freaks, I would guess that technofreaks will be with us for quite a while. Technofreaks are impatient with all constraints, especially ethical, legal, and organisational constraints. They are secure in their own knowledge (or ignorance) and extremely goal-oriented. Their goals, however, are often personal rather than organisational, and for them, the value of the work is in the exercise of their skill with these new tools.

This group thrives on technological jargon, and will invent their own if you don't provide it. They want to know all about bits, bytes, MIPSes, etc., and how to get them all for their own use. They will generally be satisfied with generalised instruction, good reference materials, and a high-powered consultancy able to answer quite detailed questions. The questions they ask, however, will often be second order questions, that arise in the course of an attempted solution to the problem they are really trying to solve. With technofreaks it is always wise to ask *why* they wish to know something.

This group will be the most difficult to bring to maturity because of their self-centrism, but for this very reason it is important that you keep track of them. And the best way to do this is through a continuous training and consultancy program. As you discover what they are trying to do, you may be able to provide the tools to enable them to do it in a constructive fashion.

The appliers of technology generally avoid the excesses of both the technophobes and the technofreaks, being willing to use technology, but not forcing their jobs out of shape to give them an excuse. They will have a tendency towards conservatism, however, that you may find a little difficult to understand. Once they know how to do something, they will see little reason to relearn it, just to use a new tool (even if it is "better" in some sense; in strong contrast to the technofreaks, for the appliers of technology, the value of the tool is in the doing of the work). Training for these people can avoid the technical detail craved by the technofreaks, but if a smattering of technical detail can help them understand certain constraints or pitfalls, they won't shy away from it. In common with technophobes, they would rather have examples and templates than explore on their own, but unlike technophobes, they will respond to suggestions on how to extend their use into related areas. They will be less imaginative than the technofreaks in their development of new applications, but perhaps more demanding in ways to link the results of different applications into a meaningful stream. This group is the largest, and, fortunately, should be the easiest to bring to maturity.

Who pays the bill?

The answer to this question, as always, is "The end users." One way or another, whether IT services are charged directly back to the users, subsidised, or funded out of overhead, whether end users get their computing and their training from the central IT function or from some other source, it is the end users who pay. Different costing schemes have different effects on the users, however. In general, if real money must be spent for a service ("real" money is money that can be spent in the real world), that service tends to be underutilised. (IT, in its maturity, must recognise this tendency, even if, in a sense, it is a demonstration of the lack of maturity of the users. Education may reduce the tendency; it will not eliminate it.) Therefore it is in the corporate best interest to see that instructional services, including both training and consultancy services, are funded indirectly, so that they are perceived as free. If a user has to pay for each phone call to a consultant, he will blunder forward on his own best guess or that of his office mate rather than make the call; the potential loss to the corporation is enormous.

If they do not already, your end users will soon have at least half the computing resource under their control; they already have all of the personnel resource. It would seem to be wise to encourage the wise -- *mature* -- use of this investment.

LAWRENCE BERKELEY LABORATORY
TECHNICAL INFORMATION DEPARTMENT
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720