Recurrent dilemmas of computer use in complex organizations

by ROB KLING and WALT SCACCHI
University of California, Irvine
Irvine, California

COMPUTER SYSTEMS AS TOOLS

Computer technology is usually spoken of as a problem-solving tool, a helpful device used to ease the burdens and expand the flexibility of information processing. In this narrow sense, computer technologies have in fact increased the capabilities of people and organizations to carry out complex calculations, manipulate large sets of data and access data from geographically remote locations.

These capabilities generate a corresponding and sometimes unexpected set of problems for many computer users. People who use computer systems for a variety of daily tasks must adjust to changes in computer systems, vie for adequate priority for their computing jobs, develop backup procedures when automated systems fail and periodically search for skilled programming staff. As a result, the very technology which was supposed to be an unobtrusive aid and time-saver can become very attention-demanding and a source of continual low-level conflicts. The “problem solving instrument” is capable of generating its own special problems.

Easing problems of computer use has been a traditional concern of computer scientists and many solutions have been suggested and tested. Most of these solutions, however, have assumed that computing is a fairly straightforward dialogue between a hypothetical user and a machine. Focus may rest on one party or another. Thus, hardware-based solutions which focus on expanding the flexibility and reliability of the machines emphasize components such as new peripheral devices, distributed computing, microprocessing, operating systems protection schemes or computer graphics. Likewise, software-based solutions which focus on easing the cognitive burdens of the user include new programming languages, data base manipulators, or more “natural” interfaces. Lastly, managerial solutions emphasize the organizational arrangements within which computer based-services are developed and provided. Involving users in systems design, for example, is often recommended to ensure that system specifications are appropriately developed.

Analysts can suggest sensible solutions to difficulties that computer users face in dealing with computing by segmenting the world into manageable chunks. Named topics such as “ease of access,” “software reliability” and “resource allocation” are well known labels for identified problems. This is the traditional “divide and conquer” strategy of the engineering disciplines and helps make complex production problems manageable. Solutions to these identified problems, however, reduce only a selected portion of the burdens faced by computer users. As computing use grows in complexity, and the number of identified “problems” and “effective solutions” increases, the likelihood that they can all be well handled by any group of service providers or instrumental users diminishes.

The routine use of computer-based services increasingly brings people in computer-using organizations into a complex set of dealings with the technology, its providers and other actors. These social relationships are both a source of service for computer users and a locus of difficulty. Factoring these relations into independent “problem areas,” each with its own technical and managerial strategies for solutions, doesn’t help a user comprehend the way in which computing is often problematic. First, no profession or service provider is usually capable of meeting all the needs and wants of its clientele. Secondly, the problematic aspects of computing arrangements often interact. Problems are best factored when their components interact weakly. In the case of computing, choices of which technology to use, who to staff it with, how to maintain it, and how to pay for it are often highly coupled. These are clearly social decisions as much as they are technical decisions. The social aspects of computer use are commonplace, but nevertheless they are poorly understood.

Our studies of computer use in a variety of settings indicate that many users often have recurrent problems in obtaining computer services smoothly. Management analysts are quick to suggest that when users have difficulties, there must be a clearly identifiable management problem which needs a systematic solution. In most of the organizations we have studied, managers and staff have developed sensible strategies for dealing with many aspects of computing; but problems still recur. It is easy to blame recurrent problems on “poor managers,” “stupid users” and “inadequate technology.” Such sentiments are too loaded with blame and faith in simple solutions (i.e., “education,” “more core”) and too short on analysis to be uniformly

From the collection of the Computer History Museum (www.computerhistory.org)
convincing. Simply identifying new "problems" and suggesting new, independent "solutions" may even add to the burdens of attention faced by computer users. We suggest a new approach to help understand why computer use is often problematic.

We find it helpful to expand the traditional view of computing from that of a "tool" to that of a "package." The tool metaphor, which is very appropriate for simple, individually controllable devices, such as hammers and pocket calculators, suggests that the item denoted may be used with few attendant problems. Of course, some tools may be more graceful, effective and reliable than others; but in most cases one can safely focus on the device to understand its use and operation.

In contrast, the package metaphor describes a technology which is something more than the physical device. In the case of computing, the package includes not only hardware and software facilities, but also a diverse set of skills, organizational units to supply and maintain computer-based services and data and sets of beliefs about what computing is good for and how it may be used efficaciously. Many of the difficulties that users face in exploiting computer-based systems lie in the way in which the technology is embedded in a complex set of social relationships.

Not only are most computer systems shared with other users, but programs and data are provided through several different social networks which often entail contact with different social groups. This complex social setting in which computing is embedded makes computing a social object, and the use of computer-based services a social act.

The primary thrust of this paper is to identify the recurrent aspects of the social world of computer users which are problematic for people who use computing to serve other ends. We have expanded our conception of computing as a potentially problematic "tool" to computing as a social object. We will now explore some more specific consequences that this expansion reveals. We would caution that while we list a set of issues which are problematic for computer users and computer specialists, and advance some hypotheses as to their relative and absolute costs and importance, we intend this discussion to be an introduction to the bundle of issues which warrant further investigation, articulation and conceptualization.

THE SOCIAL CHARACTER OF COMPUTING

Our analyses of computer use are based upon several empirical observations and theoretical claims:

1. Many people (instrumental users) who use computing hope it will help them be more effective in their work. The substance of that work may have little connection with computing; computer use is a means to further some other end.

2. In many important situations in which computing is used there may be many different people who are interested in utilizing the same computer-based data or reports. These people can have different understandings as to the capabilities of computing and indicate different interests in the uses of computer-based analyses.

3. Much modern computing and most important automated information systems are supported in settings in which several specialized groups provide the requisite computing services and data. Automated information systems serve managers and organizational people who have little time or skill to carry out the full range of computational tasks to support their data use. Even skilled programmers rarely design, implement, test and maintain all the software they use while carrying out their work.

4. Users of computer-based services frequently report an array of difficulties in computer use. Complaints usually focus upon aspects of computer use which are byproducts of the social arrangements in which computer-based systems are conceptualized, developed, provided and maintained. These problems rarely focus upon computing hardware, except when users believe there is too little of it or when some party allegedly chose less suitable equipment than might be available in the market.

5. Computer-based services and information processing tasks are organized in vast array of distinctly different arrangements within and between organizations. Smooth computing use often entails the cooperation of distinct organizational groups and interests.

6. We view organizations as patterned arenas for conflicting and cooperating interests. We note there are often conflicts between the interests of participants who identify primarily with computing as their profession or career interest, and those who identify with some other social world in which computer use is primarily an instrumentality. These extremes are, of course, simplified since many participants align themselves as specialists who mix computing and other substantive interests. But the grounds for conflict of interests remain similar.

These observations encourage us to view much of computer use as a complex social phenomenon in which hardware and software plays an essential, but partial role. In fact, computer use can be expected to be particularly problematic as the milieu in which it is embedded increases in social complexity.

ISSUES IN INSTRUMENTAL COMPUTER USE

Computing services are produced and consumed in work settings in which the participants take on specialized roles. The demands that instrumental users and computer specialists make upon each other and of the technology hinge, in part, on their understandings of the appropriate role, capabilities and limitations of computing. Typically, relations between service providers and their clients is problematic. Few service providers can meet all the wants of their clients or of the organizational participants to whom they are ac-
countable. Few clients have sufficient skill and interest to deal with technically skilled service providers on their own terms.

Application development changes procedures and processes for users at various intervals which may be either relatively benign or disruptive. When instrumental users rely upon automated data systems, ensuring that the data provided is of high quality (i.e., accurate and timely) is particularly sensitive. Part of the social interaction around computing involves establishing and maintaining control over the various computing resources within the organization.

Both specialists and users depend on the current state of software development practices to help construct reliable programs which are easy to operate and maintain. Similarly, computing creates special demands for the time and attention of users. The social aspects of computer work and computer use play a large role in shaping each computing milieu, as does the particular technology in use.

In Table I, we list the array of representative issues which we have clustered under the categories emphasized in the preceding three paragraphs. This set of issues has been selected because they appear problematic to instrumental users in studies we conducted in a variety of settings. Some of the specific issues indicated in Table I are important to computer users in many settings; others occur infrequently. Most of these issues should be easily recognizable since they are common in settings where there is extensive computer use. These issues are briefly examined in Appendix A to indicate how each one is a byproduct of the social elements of the computing package and how it can effect the quality of computer-based services.

These issues do not exhaust those raised by the social nature of computing. But they do represent those social aspects of computing which strongly influence the patterns of computer use adopted by instrumental users. The relative importance of any of these issues is also dependent on the organizational setting where computing occurs.

### TABLE I.—Common Issues in Instrumental Computer Use

<table>
<thead>
<tr>
<th>The Work Setting of Computer Use</th>
<th>Data Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The concepts users and computer specialists have of their own work and the role of computing in it.</td>
<td>1. Collecting input data.</td>
</tr>
<tr>
<td>2. The mutual perceptions of computer specialists and users.</td>
<td>2. Ensuring the correctness of processed data and analyses.</td>
</tr>
<tr>
<td>3. Differing responsibilities among computer specialists.</td>
<td>Control Over Computing</td>
</tr>
<tr>
<td>4. Doing a &quot;good job&quot; and being rewarded for it.</td>
<td>1. Control over the technology.</td>
</tr>
<tr>
<td>5. Maintaining career mobility.</td>
<td>2. Access to and control over expertise.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Understanding the Role and Capabilities of Computing</th>
<th>3. Controlling the kinds of demands made by users.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Learning about computing—what computers are good for, how their particular machine might be used, etc.</td>
<td>4. The &quot;values&quot; sought after by those individuals and organizations which promote applications development.</td>
</tr>
<tr>
<td>2. Getting a computational task successfully completed.</td>
<td>5. Developing and maintaining political support within the organization.</td>
</tr>
<tr>
<td>3. Dealing with computing/systems jargon.</td>
<td>Software Development Practices</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Changes in Computing Arrangements</th>
<th>2. Program testing.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Locis of change.</td>
<td>3. Software maintenance.</td>
</tr>
<tr>
<td>2. Scope and rate of change.</td>
<td>4. Software documentation.</td>
</tr>
<tr>
<td>3. Formalizing change procedures.</td>
<td></td>
</tr>
</tbody>
</table>

### Attention

1. The kinds of attention demanded by computing.
2. The precision and detail demanded by computing.

The actual difficulties experienced in using computing depend upon the interplay between both technical and organizational arrangements. Consider, for example, the different impacts of data base management systems (DBMS) on the time to produce a program for a user in scientific and commercial settings.

Computer specialists may assume that a scientist utilizing a DBMS will either carry out his own programming or employ a skilled research assistant who is under his supervision. This is a result of the work organization of scientific laboratories in which each research team has dedicated research assistants to help carry out a variety of laboratory chores including data collection, reduction and analysis. If the scientist desires to change schedules or priorities in his use of the DBMS, he normally faces no bottlenecks in the process except the limitation on his own or assistant's time. Since he can regulate these alterations of priority, he is at most buffered by one queue from access to programming.

A different situation faces the instrumental computer user in a commercial firm. In commercial firms, it is rare for staff to have their own programming assistants. Programmers are usually centralized in a pool, even in user departments and scheduled through a supervisor. The commercial user may thus be further buffered from the access to computing. He may have to negotiate with a supervisor, a special committee or a review board to achieve changes in schedules or priorities in dealing with a DBMS. Each of these parties has a separate queue of requests and demands with their attendant delays. Each such queue creates additional delays for the commercial user in gaining access to programming assistance. In practice, a person may wait much longer to get on the queue of a programmer than it takes to do the work.

Even if a DBMS reduces the time required for a programmer to write a given program, the time it takes for users to
get a given computing task completed depends upon orga-
nizational arrangements. This example illustrates the way in
which the social setting of computer use may influence users
more than the technology in use.

STRATEGIES AND RESOURCES

The issues identified in this paper are representative of
those that arise for many instrumental users and computer
specialists in their daily encounters with computing. Im-
proving the grace or ease with which computing is used
hinges on coming to grips with these issues. This requires
recognizing computing as a social object as much as it de-
pends upon developing new software and new hardware. In
addition, a major impact on groups using computing is the
increased attention to information processing—its manage-
ment and conflicts—that negotiating these issues demands.
People’s time, skills and organizational resources are in-
volved in attending to these negotiations. The negotiation
costs, in time, money, skills, foresaken opportunities, and
sentiment borne by instrumental users may become a sub-
stantial fraction (if not the largest) of the cost of a system
during its life cycle.

Computer specialists have been sensitive to some of the
difficulties of computer use raised here; after all, they are
commonplace. And computer scientists have been particu-
larly adept at providing technical solutions for some of these
difficulties. Generally, those technologies that diminish the
’social size’ of the computing package by decoupling in-
strumental users from some of the groups upon which they
depend may alleviate some of the burdens of computing.
Thus, acquiring a minicomputer may insulate a group of
instrumental users from demands for machine resources
made by other groups. However, it doesn’t diminish the
difficulties of managing data and may even increase the
difficulties instrumental users face in managing skilled staff.

‘Turnkey’ installation of applications and hardware may
reduce the instability of computing development. However,
other technical improvements are more problematic from
the perspective developed here. While advocates of data
base management systems have stated objectives of making the
development of ad hoc analyses easier for instrumental
users,25-29 the social complexity of the computing milieu
should increase since new specialists (such as data base
administrators) are often employed. It is empirically open
whether the overall environment of data base management
is easier or more difficult for instrumental users to negotiate.
Similarly, software engineers often propose that develop-
ment aids such as test data generators would help insure the
correctness of programs. From our perspective, a test data
generator, however carefully crafted, is another package
subject to the recurrent social histories of computing pack-
age.

Management and social analysts who identify difficulties
of computing in the social milieu often propose organiza-
tional reforms such as new pricing schemes or design disci-
plines that emphasize user involvement.16,21,27 Such stra-
tegies often resolve particular dilemmas of computer use in
a specific setting, but they do not deal directly with the
large, diffuse social elements that pervade the computing
package.

Our own field work in several large private firms and
research laboratories indicates that effective strategies often
to undesirable commitments of organizational resources.
Chains of liaisons between instrumental users and comput-
ing service providers facilitate multiple lines of communi-
cation and smooth tensions between conflicting groups. Reg-
ular meetings and redundant forms of communication ease
coordination and minimize the likelihood of major slippages
between the service providers and their clients.

Technology-based strategies often require large resource
commitments. We have seen, for example, one engineering
firm which uses a large-scale computer for production ap-
lications, and a similarly large computer devoted solely to
software development so that the work of many instrumental
users.22 However, these
difficulties of computer use presented in this paper, we draw the following conclu-
sions:

1. The computing tool metaphor displaces attention from
the social dilemmas of computing by tacitly identifying
advances in computing with advances in the technical

CONCLUSION

Much of our account has focused upon the problems at-
tenant in routine computer use. This is not because we
believe that computing is a wholly troublesome technology.
On the contrary, we believe that computer use often in-
creases the information processing effectiveness and eases
the work of many instrumental users.28 However, these
gains often do not come gracefully or easily. Computer use
is most troublesome when the necessary social resources
(such as technical expertise, demands for time and attention,
staff sentiment and control over computing services) are
slighted or ignored when new computing arrangements are
"budget strategies" to
be the rule rather than the exception. Given constrained
resources, some interests will be better served than others
and some parties should be expected to face computing
dilemmas routinely. The empirical prediction would be that
any problem (such as data quality, response time, appropri-
ate consulting or adequate documentation) should be trou-
blesome for some minority of instrumental users in even the
best managed setting of computer use.22

In summary, technology-based strategies often miss major
portions of the computing package that include important
social relations and contingencies. In addition, the best mix
of technology-based strategies and socially-oriented strate-
gies for graceful computing can consume large resources,
time and money. Since most computer using groups have
limited resources, one should expect “budget strategies” to
the exception. Given constrained
resources, some interests will be better served than others
and some parties should be expected to face computing
dilemmas routinely. The empirical prediction would be that
any problem (such as data quality, response time, appropri-
ate consulting or adequate documentation) should be trou-
blesome for some minority of instrumental users in even the
best managed setting of computer use.22
sophistication of the equipment used. Moreover, many of the attendant difficulties in computing are not well predicted or understood by employing the tool metaphor.

2. Problems of computing vary with the particular computing technology in use and the organizational arrangements through which computing services are produced and used. Hardware reliability is usually more salient in on-line systems than with batch systems. Allocation of machines and staff are typically most contentious when control over each resource is centralized. Staff highly sophisticated in computing may provide the best technical assistance, but they are also the most difficult to interest in routine applications.

3. Many of the problems experienced by computer users develop from their relationships within the "computing world." The computing world is highly differentiated into specialty interests and organized to routinize the movement of innovations from producers, through service providers to instrumental users. Instrumental users face markedly more complex issues when they split their computing activities across equipment supplied from different "vendor worlds." Also, they often have little control over the pace at which small enhancements or alterations are made in supporting software supplied by groups outside their organization.

4. As technical advances in computer hardware and software simplify the technical problems of computer use faced by users, the social problems of computer use will become relatively dominant. Each of these problems has associated costs. These costs are poorly understood and have yet to appear in the figures cited for total systems costs.

5. The social elements of computing are typically underestimated in proposals for new computing arrangements. Social resources such as time, attention, skills, information and inclination can be costly to acquire, utilize and maintain, but discounting their role in computing results in displaced organizational costs. For example, expert consultants and good system training aids are costly to provide. But when instrumental users cannot obtain needed assistance, they recurrently find computing use to be troublesome and uncertain.

6. The package view implies that successful computer use depends on the organizational distribution of social and technological resources and how they are allocated or acquired. Successful computer-using organizations balance their technological investments with explicit investments in the social elements of the computing package.

7. Currently, there are no simple or uniform solutions. Alternative computing arrangements which are proposed to solve certain individual problems can exacerbate or manifest others if the social character of computing is disregarded.

Computing is a problematic technology for many instrumental users, in part, because it raises so many social issues which continually demand attention. We suggest that instrumental users and specialists alike use the list of issues presented in Table I as a diagnostic guide for assessing the impact of existing or proposed computing arrangements. As a checklist, Table I can help an analyst decide which activities a new "solution" may alter, and which it may leave untouched. Table I can also help an analyst make explicit the rich set of social features which characterize the social milieu of complex organizations.

ACKNOWLEDGMENT

Elihu Gerson, Sharon Price, and Harold Sackman provided helpful comments on an earlier draft of this paper. Philip Crabtree helped develop some of the ideas presented here. This work was supported by NSF Grant MCS77-20831.

APPENDIX A—COMMON ISSUES IN INSTRUMENTAL COMPUTER USE

The work setting of computer use

The concepts users and computer specialists have of their own work and the role of computing in it. Specialists and users have different concepts of how central computing is, should and could be to the successful performance of their jobs. To specialists, computing can be everything. Instrumental users, however, often view computers simply as a means to achieve some other ends. This difference of focus has substantial repercussions for the amount of effort people of each orientation are willing to spend learning and adapting to new computing system developments.

The mutual perceptions of computer specialists and users. Specialists can influence the involvement of users in the computing process. Shared perceptions may be important to the specialist in determining how users should be educated or to what extent users should be involved in the design, implementation and maintenance of particular systems.

Differing responsibilities among computer specialists. As an organizational unit grows and expands, the jobs within it often become more narrowly defined and specialized. The resulting division of responsibilities and skills may increase the difficulties faced by clients of the unit when they seek a service which requires several specialists. For example, an instrumental user may find that to change an inquiry program, he or she must coordinate efforts with those of a programmer, a systems analyst, a data base manager and a teleprocessing specialist. Increasing the technical sophistication of a system often leads users to interactions with more specialists.

Doing a "good job" and being rewarded for it. People often differ on which aspects of a job are important for satisfactory performance. Some programmers emphasize satisfying user demands, while other programmers emphasize elegant code.

Despite these individual interpretations of what constitutes doing a "good job," the organizational structure may impose a reward system on specialists which emphasizes...
different activities. The rewards may be for meeting schedules, for the number of coding lines produced or for getting to work on time. Whatever reward system exists in an organization for computing specialists, it may conflict with what specialists perceive to be important measures of job performance.21

Maintaining career mobility. Specialists appear different than any other employees in being concerned about job security and career development. Specialists may feel that a strong position in the marketplace depends on one’s experience with the latest technological innovations. Consequently, specialists may influence their organization to continually acquire state-of-the-art hardware and software packages.23

Understanding the capabilities of computing

Learning about computing—What computers are good for, how their particular machine might be used, etc. Beliefs about the appropriate role and capabilities of computing vary considerably. Those who work closely with the technology often view computing as a special-purpose device which is best suited for applications something like their own. Thus accountants often view computers as “accounting engines,” while urban planners may view them as statistical calculators.

Coupled with beliefs about appropriate tasks for automation are beliefs about the ease of applying computing. Computer specialists often view the technology as speedy and convenient. However, programmers (like planners, designers, managers and other professionals), can underestimate the time required to develop and implement new projects.

Getting a computational task successfully completed. When a problem can be solved with the existing computing system, users may find themselves facing a procrustean software system.20 Rigid system designs add to the complexity which users must overcome to compute a solution to their problem. In theory, computing may be both technically and organizationally complex. In fact, it is also complicated.* Most software packages, however simple or complex, usually have idiosyncratic conventions** which arise from problems in implementation, compatibility with odd features of related systems, or simply through “poor” design. Nevertheless, an instrumental user must master and remember these conventions to utilize a software system.

In addition, the elapsed time to complete a computational task, from the point of view of an instrumental user, begins when the task is conceived and ends when the computed job is translated into a usable form. This time frame is larger than that of the computer specialist who counts from the time that a task is well specified until a product is delivered to the user. And it is still longer than the time to complete a computational job as viewed by the computer operators. This usually equals the time to complete a job once it is being executed by a digital computer. Despite these obvious observations, the time to complete computing tasks are usually conceptualized in time frames closer to those of machine execution than to those of instrumental users.

Dealing with computing/systems jargon. Specialized languages enable work groups to communicate about their work compactly and to maintain a definition of “insiders” and “outsiders.” “Jargon” is thus an inevitable element of workplace in specialized occupations. Smooth expert-client relations in computing milieus require that either one participant know the technical vocabularies and rationales of both computing and the occupational world to which it is applied, or that one of the parties be skilled at developing communicative bridges between computing and another world of discourse. If both parties possess either skill, so much the better.

Purposive use of jargon enables an actor to structure situations to her or his advantage by “snowing” the other parties in an encounter. When the legitimacy or competence of a computer specialist is brought into question, confident explanations couched in complex technical rationalities are difficult for most people to penetrate. Purposive use of jargon helps a specialist save face and protect his autonomy.

Getting adequate documentation for computer-based systems. Both computer users and specialists rely upon a variety of documents to learn the capabilities of and precise incantations for using particular system features. Different users of a given system may desire either a tutorial manual or a reference manual, although both rarely co-exist for any computer-based system. In fact, one dominant feature of computer settings is the extent to which participants depend upon clear and accurate documents to select, use and maintain computer systems, and the relative paucity of appropriate high-quality documents. Documents, like any other computing product, are produced within a social order of computer specialists, service providers, and clients. The difficulties of documentation are more those of the priorities within the computing world rather than the technical difficulties of writing.30

Changing computing arrangements

Locii of change. Most computer applications evolve gradually. However, changing an application often results in altering the routine procedures for many different people who use it. New features are usually negotiated between some mix of instrumental users, computer specialists, important actors in the computing organization, computer vendors, and outside consultants.

Computing personnel often have more requests, demands and self-initiated ideas for changes than their staffing per-
mits. Thus, they can usually select certain alterations from the larger set of requested or required alterations. While many changes in computer applications or their supporting systems are requested or "needed" by some users, certain users appear better served than others. In addition, most users must expend personal and organizational resources to ensure that changes which they desire are actually implemented. The actual dynamics of these negotiations, the resources they consume and their repercussions for both computer users and computer specialists are poorly understood.

**Scope and rate of change.** Changes in the computing milieu vary in frequency and scope. While it is easy to assume that low rates of change are easier for users to adapt to, that hypothesis is oversimplified. Infrequent changes of wide scope, such as changing the formats for large sets of data, may disrupt a class of users regardless of frequency. Upward compatible features which are transparent to most users may be introduced into many systems and processors with relative impunity for most users.

On the other hand, certain users often seek specific changes in both applications and support software. However, in shared systems, changes developed for one party are typically imposed upon all users of the same computational resources. Many technical changes that benefit one party may benefit others as well. However, there are also common conflicts between the technical needs of different users. It is an empirically open question as to how frequently technical changes are either "pareto optimal" or indicate a redistribution of computational resources. Thus, the advocacy and implementation of changes has a strong political content above and beyond the resources required to implement the change.

**Formalizing change procedures.** Large software systems are often used to support organizational activities. Since these systems operate in a production environment, any alteration to such a system is usually scrutinized to assure that it doesn’t cause disastrous effects. Many organizations have thus instituted a series of bureaucratic procedures to be followed prior to the actual alteration of a system. These procedures can be cumbersome in certain organizational structures.

The ability of an organization group to get a particular set of changes implemented may require interaction with, and the approval of, a number of intervening individuals or committees. If there are a large number of system change requests pending, then some prioritization scheme may exist. The group seeking system changes may now have to rely on its members’ negotiating skills to assure a suitable priority.

**Data quality**

**Collecting input data.** Many contingencies structure the situations in which one party collects data about the activities of a second group from a third group for use by a fourth group. Some extreme situations include those in which (a) the groups are all the same or (b) all groups are aware of each other and share information with mutual consent and are all jointly concerned that the data be accurate. The latter case might occur with bank records, for example. In cases of high mutual commitment, data capture may be smooth and subject primarily to errors in data entry.

However, in some important situations conflicts of interest or priority can arise among the various groups. If the data is to be used to assist the fourth party to control some activities of the data subjects (as in tax reports) there is some incentive for incomplete or inaccurate reporting. When several organizations share information systems, providing high quality and complete information may be more important to some participants than to others. Thus the quality of data collected is influenced by the patterns of interest cooperation within the social order surrounding the information systems.

**Ensuring the correctness of processed data.** It is common to believe that once data is accurately captured by a computer-based system it will remain accurate. There are at least two conditions under which this assumption can be problematic. Sometimes data is aggregated or reorganized to be used in an analysis. As the complexity and number of the data manipulation steps increases, programmers, operators or the application system itself may introduce difficulty to detect errors in the transformed data set. Since data do not reorganize themselves in useful ways without personal intervention, data analysts are an essential part of many policy analysis units, survey research centers, etc. Secondly, in some systems which are shared by many users, particularly simulations, important parameters may be changed by one party without the cognizance of other users. Digital computers are particularly useful as calculating engines when the computations are too complex, tedious, or time-consuming for hand calculation. However, it is just in such cases that verifying the validity (or stability) of the results obtained is the most difficult. While such events are rare, their dynamics are instructive.

**Control over computing**

**Control over the technology.** Maintaining effective control over computing resources is a central issue for many computer users in an organization. In addition, some higher-level administrators who are not computer users, simply view computing as an expensive line-item to be kept in check.

Since computing and information are rich organizational resources, issues over contention for control are naturally commonplace. Like other social aspects of computing, negotiations over control of specific computing resources (e.g. data, programmers, budgets, I/O devices) take time and absorb organizational resources.

**Access to and control over expertise.** Computing is a complex process. In spite of its complexity, its use by a variety of people is becoming widespread. Many people do not take the time or interest to learn a great deal about it. Therefore, they must rely on others to help utilize computing effectively and to handle problems and unanticipated situations. There is increasing evidence that when users have easy access to expert assistance, the computer-based systems are better accepted than in those situations where access is more difficult.
Controlling the kinds of demands made by users. Whenever a personal service is provided, the stage is set of its consumers and providers to continually negotiate the kinds of service each would most prefer. In this way, computing is little different from other personal services such as legal advice, financial counseling, or medical care.

Computer specialists develop strategies for managing the behavior of their clients to help serve their own ends and make their organizational life tractable. Since they usually work as salaried employees with little freedom to negotiate a higher wage for difficult projects or those that incur an unacceptable level of dirty work, the strategies usually entail claims about organizational contingencies. Users may be told their requests are more expensive to fulfill, will take longer time to complete or entail unexpected technical complexity (such as system redesigns) to help displace less desirable work. Since computer specialists often have a relative monopoly on the expertise essential for judging the complexity of different requests, specialists’ work-moderating strategies are difficult for instrumental users to easily counter.

The “values” sought by those individuals and organizations which promote applications development. Often computing systems are developed and installed when a specific person or small group of individuals actively promotes computing within an organization. New computer applications are usually costly; promoters who want resources allocated to their project must often first obtain sanctions from other organizational members. Since different actors become involved in acquiring computing resources, computing will often serve many ends. For example, some actors may be seeking to enhance their administrative control, others seeking to cut costs, still others may be seeking to make their jobs easier or more interesting. Few applications can be designed to serve many different interests well. Some users of computing often face difficulties which derive from the way in which their system is “optimized” to serve the interests of some other group.

Political support within the organization. Politics deals with the allocation of goods, services, symbols and values. The distribution of computing resources is often the focus of conflicts over budgets, staff and domain. This is not incidental. Rather, it is an intrinsic aspect of computer use. To the extent computing resources are valued by different actors in an organization, they will seek access to them. The resulting contention with its usual conflicts, bargaining and subterfuges is similar to other kinds of organizational politics.

Some actors seek control over computing resources simply because it provides a relatively large, growing staff and consequently a growing budget. There is also some evidence that overall computing arrangements can be more strongly influenced by the political access of key actors than by the technical soundness of their preferences.

Software development practices

Programming and design practices. Since the development of software has become a major expense of computing for most organizations, considerable attention has been focused on improving software design and programming productivity. New techniques and tools have been developed to assist specialists with their various tasks. Structured programming is currently emphasized to aid specialists, as well as Chief Programmer Teams and automated design aids.

While these techniques and aids may be beneficial for the organization, they may be problematic and disruptive for specialists. They may actually make the specialists’ jobs more difficult and attention-demanding. They may create changes which are frustrating for specialists accustomed to previously established procedures.

Most modern programming practices and tools are yet to be widely adopted in computing settings outside of where they were developed. Reasons for this are unclear, but we suspect that organizational contingencies (such as meeting schedule deadlines, budgetary constraints or personnel training costs) in a computing setting tend to shape the adoption and incorporation of such tools and techniques.

Program testing. In theory, one would like to be able to automatically generate a sufficient set of test data necessary to demonstrate the probable correctness of a robust class of programs. However, for programs of moderate complexity, the set of test data to exercise all paths through a program is infeasibly large. Nevertheless, some promising research is proceeding on various schemes to automate tests for special program conditions. In contrast to the research on new tools for program testing, the state of current practice does not rely upon much automation at all. Test data, for example, are usually selected manually by a programmer or a knowledgeable user.

Software maintenance. Maintenance is often considered to be everything that happens to software after user acceptance. Maintenance can range from “bug fixes” through complete redesign and redevelopment of a delivered system. Often, the people who develop the system are not the same as those who maintain it. Given that different specialists are involved in system development and maintenance, finding those people (users or specialists) with dependable understandings of a system operation can be quite salient in determining the ease, timeliness and reliable execution of maintenance tasks.

Most programming work is in maintenance, not development. However, maintenance entails ongoing interaction between users, programmers, managers, vendor representatives, etc. While current software system life-cycle costs reflect the high cost of maintenance, the available figures do not distinguish the costs of program alteration work from the time, skill and attention required by specialists to successfully interact with those people requesting alterations. The extent to which these interactions are negotiated and completed with ease or difficulty, may better account for the variation in minimizing or exacerbating the costs of “routine” maintenance tasks.

Software documentation. Adequate and up-to-date software documentation is continually a weak feature of most software systems. Poor documentation is not usually a result of some software development practice. We note that while
many software system manuals can be measured in inches, their adequacy and currency vary. However, reasons for the variable quality of documentation appear not to be due to the unavailability of suitable documentation support aids or deficient programmer practices. Rather, updating documentation demands time, skills in clear and concise writing and attention. Given that specialists face some number of competing demands for their services, their ability or desire to maintain documentation competes with other work demands whose completion may be more highly rewarded.

Attention

The kinds of attention demanded by computing. Computing may appear to some users and specialists as a technology that requires a person learn a great deal to effectively utilize it. Many users must (or are at least led to believe they must) use the computer efficiently because it is a scarce resource. However, the time required by a user to prepare and successfully execute (after ‘debugging’ runs) efficient programs often displaces any net savings in terms of completing the task at hand. Concern for minimal computer resource usage versus concern for minimizing the time to complete a work task often lead to conflicting demands for the user’s attention.

The precision and detail demanded by computing. As a tool, the computer is a fairly exacting device. It demands that procedures be followed explicitly. It does not allow loose or **ad hoc** procedures in handling transactions as might exist in a manual or more informal information system.

At times, users complain that their jobs are actually more difficult or less interesting with computing than they had been previous to computing. Specialists complain that it takes a special person, like a ‘hacker,’ to be truly satisfied with the detail demanded by systems and application programming.

REFERENCES


From the collection of the Computer History Museum (www.computerhistory.org)