

**BARRIERS TO THE ADOPTION
OF
APPLICATION SOFTWARE PACKAGES**

Pamela HB Gross
Drexel Burnham Lambert Incorporated

and

Michael J. Ginzberg
New York University

December 1983
Revised January 1984

Center for Research on Information Systems
Computer Applications and Information Systems Area
Graduate School of Business Administration
New York University

Working Paper Series

CRIS #64

GBA #84-22(CR)

Abstract

A two-stage study was conducted of potential software application package users. The first stage defined 38 issues which were viewed as potential obstacles to the purchase and implementation of applications packages. In the second stage, the relative importance of these 38 issues was quantified. The results suggest that the key obstacle to wider usage of applications packages is uncertainty. Various steps which package developers and package users can take to reduce uncertainty are suggested.

Short title: Adoption of Software Packages

Keywords: applications software, software packages, system development process, user-vendor relationships

Introduction

In the past several years, software has emerged as the critical issue in the computer industry. Hardware, long the dominant concern of DP managers, no longer holds that special position. This change is evident in the spending patterns for data processing. In 1977, hardware (including mainframes, minicomputers, peripherals and terminals) accounted for 76% of DP spending while software and services accounted for 18%.¹ By 1979, hardware had dropped to 70% of DP spending while software had risen to 26%. During this two year period, total DP spending rose by 47% (from \$31.1 billion to \$45.7 billion), and the actual dollar spending for software and services increased by 108% while spending for hardware increased 36%. There is no reason to believe that the pattern has changed since 1979.

The basic reasons for this shift are well known. Technological advances have drastically reduced the cost of hardware, making raw computing power increasingly affordable. The development of powerful mini- and, more recently, microcomputers have placed the computer within the reach of broad new classes of users. These new users have expanded computer usage well beyond traditional accounting and operations support applications which have been the primary applications of computers in the past. Wider distribution of computing power has created a demand for more software. It has been estimated that

for every computer purchased, 150 to 400 programs are required [2]. The 80/20 ratio of hardware to software expense common in the 1960's is reversing itself in the 1980's.

The increasing demand for software has not been easy to fulfill. Many companies are reporting large backlogs of requests for applications (see e.g., [11]); some are even measuring them in "man-centuries." Worse yet, by one report [1] there is an even larger "invisible backlog" of applications users would like to have, but because of existing large backlogs have not yet even requested. Estimates by the Bureau of Labor Statistics suggest that there is a 10% deficit of experienced personnel to fill existing positions which will persist at least until 1990 (see [7]).

A number of approaches to solving this "software crisis," to filling the "applications gap," have been suggested. Included are (1) the use of very high level (4th generation) languages, applications generators, etc., (2) more end user programming, and (3) the use of application packages. The focus of this paper is on the last of these three. The two major advantages cited for application packages (compared to in-house development) are cost and timing. The cost of a package is normally much less than that of a comparable system developed in-house, sometimes running as low as 10% of the cost of in-house development. Further, the

time required to acquire and install a package is usually only a fraction of that required for developing a comparable system in-house (see [10]).

The advantages of packages have not been ignored. The software products component of the computer services industry grew by 30.0% in 1979 and 44.4% in 1980 (see [3], [4]). This growth is expected to remain at an annual rate of about 29% through 1985 [2]. The great preponderance of programs (69%), however, continue to be developed by already over-committed in-house staffs [9]. Apparently the products and services currently offered by application package vendors are not fully satisfying their potential customers.

The purpose of this paper is to explore the reasons application software packages are not used more widely than is currently the case. Through interviews and the administration of a survey questionnaire, we have developed a picture of the problems with packages from the user's perspective. From this base, we are able to recommend steps that can be taken by both users and vendors to increase the value and usability of applications software packages.

Methodology

This study consisted of two phases. In the first stage a small number of exploratory interviews were performed in order to develop an initial list of the problems users face in acquiring and installing application software packages. The second stage involved administration of a questionnaire to a larger sample of current and potential package users in order to gauge the relative importance of the problems identified in the first stage.

The 1st Stage: Interviews

Twenty-one in-depth interviews were conducted in 11 organizations in order to gain an understanding of their software acquisition practices. One additional interview was conducted with a software package vendor to confirm that major decision areas were not being ignored. We believe these organizations are representative of medium to large non-governmental computer users, though they are somewhat heavily weighted with service organizations. All firms included in the sample were experienced users of computer technology with applications beyond the full range of financial and accounting functions, and all had a full staff representing data processing and information systems. Brief descriptions of the 11 organizations are included in Exhibit 1.

Insert Exhibit 1 about here.

The interviews were unstructured but focused on the respondents' perceptions of four key areas: (1) an overview of the organization's software development and acquisition processes; (2) the respondent's specific area of responsibility and its interfaces with the rest of the organization; (3) major problem areas in software package acquisition; and (4) possible solutions to some of the current problems with packages. Whenever possible, multiple interviews were conducted in an organization. This provided a broader and hopefully more accurate more accurate, picture of software acquisition practices, by incorporating multiple perspectives, e.g., of end users and DP departments. Each interview lasted approximately 90 minutes.

The 2nd Stage: Questionnaires

The initial interviews uncovered 38 problems relating to application software packages. The second stage sought to quantify these results, i.e, to assess the relative importance of these 38 problems. Questionnaires were mailed to 101 respondents pre-screened by telephone. The pre-screening was done to assure that respondents were both appropriate and willing to participate in the study. 61 questionnaires were returned, of which 55 were

usable. The final sample of 55 respondents represents 32 organizations in 11 broad industry groups. The distribution of respondents by firm and industry is shown in Exhibit 2.

Insert Exhibit 2 about here.

The questionnaire included brief statements of the 38 problems uncovered in the initial interviews. Respondents rated each statement on a 5-point scale labeled: "Not a problem", "A minor problem", "Somewhat of a problem", "An important problem", and "A crucial problem." The statements were clustered, though not labeled, in the following broad problem areas: product/package content, modification of product, internal organizational factors, and vendor relationships. Respondents were also asked to provide their title, role in the package acquisition process (i.e., technical, financial, end user or a combination), and the percent of current application software needs being satisfied by purchasing software packages.

The questionnaire was pre-tested with several respondents for clarity of form and content as well as potential bias from question order or fatigue. While minimal adjustments were suggested and made, there was no indication that either length or question order presented any problem.

Results

Issues Suggested by Initial Interviews

Numerous potential problems were suggested by the initial interviews. In order to have a manageable list of problems for further investigation, it was necessary to pare down the initial list. Many of the problems suggested appeared to be company specific, situation specific, or isolated events without a high probability of recurrence. The 38 problems chosen for inclusion in the questionnaire study all had one or more of the following characteristics: (1) it was identified as a problem in several interviews; (2) it was a sub-set of a larger, more complex, and frequently mentioned problem; or (3) it represented a common thread among several frequently mentioned, related problems.

In the following paragraphs, each of the 38 problems is briefly introduced and the circumstances under which it may become more or less severe are discussed. The problem statements are grouped into the four broad classes outlined earlier.

Problems related to product characteristics

1. Source code is not usually available.

The conflicting needs of vendors and buyers come out quite clearly here. Vendors are reluctant to provide source code because its release reduces the vendor's ability both to protect its proprietary rights and to support the product. Buyers, on the other hand, want access to source code, both to protect their

investment should the vendor go out of business (or otherwise stop supporting the product) and so that they can modify or customize the package. Not all buyers want to modify packages they purchase. However, access to source code to enable modification is likely to be desired for larger and more complex packages, by buyers who are technically sophisticated, or if the buyer perceives his needs as changing in either the near or longer term.

2. Similar products are difficult to compare due to a lack of industry standard.

The number and range of participants in the application software industry is staggering. Each supplier provides its own mix of product and service. There are no standards for the components, and package comparisons are difficult to make because of the different mix of attributes that each provides. Having standards for service, product characteristics, etc. would make comparisons much easier.

3. The trade-offs present in either a 'pure' buy or a buy with modifications decision make the choice difficult.

In an ideal world, package buyers would make no modifications to the software they buy. Modifications take time, consume resources, and create a maintenance burden. As the scope of computerization broadens, however, the required applications become more and more company specific. Thus, the chance that a package exists which meets all the specifications is remote. The

task is then to identify the strengths and weaknesses of the package, compare them to stated objectives, and determine whether the difference is a sufficient deterrent to buying the package 'as is.'

4. There is an irreducible trade-off among the factors of ease-of-use, flexibility and overhead.

Optimizing any one of these factors usually results in reducing one or both of the others. But, while these trade-offs are present, they are often not considered because of the difficulty of quantifying them. Interestingly, programs developed in-house are often 'leaner' and more specialized than those purchased; implicitly, the choice made is to minimize overhead, even at the cost of 'friendliness' or flexibility.

5. There is too great a dependence on the vendor's evaluation of packages and not enough third party or objective evaluation.

One of the most widely used sources of package evaluation is the 'testimony' of current users. These pre-selected references are likely to be a biased sample, so other sources are also used. Several industry publications provide evaluations of packages, but these seldom provide the depth or breadth of detail that the decision warrants. And, of course, these evaluations cannot consider the specifics of the prospective buyer's situation.

6. Available packages do not adequately reflect my industry.

As the range of computer applications broadens and moves away from accounting systems, the peculiar needs of different industries and organizations becomes more apparent. The financial institutions interviewed felt particularly strongly that currently available packages were inappropriate to their needs.

7. My needs are too unique to be adequately represented in available packages.

This opinion was voiced in the majority of interviews, regardless of organization size or industry. Generally, it related to the issue of size; however, it also reflected a perception of doing things differently, a feeling of uniqueness. Whatever its basis, it appeared that it might be a self-fulfilling prophecy: packages are too restrictive, thus the in-house staff had to either develop a system from scratch or be involved in major modifications to an existing package.

8. One of the most difficult problems is testing a package for functional specifications.

All possible conditions the program may be required to cover cannot adequately be pre-tested because the user is not able to envision or forecast his vast array of long term needs. In addition, since the user is often not present during testing, it is a matter of guesswork for the DP person to assess ease-of-use or flexibility.

9. **One of the most difficult problems is testing a package for efficiency.**

Assessing efficiency is difficult because of the great variety of hardware/software environments in which a package might be run. This is a technical problem, and the continually improving cost/performance of hardware will likely diminish its importance.

10. **One of the most difficult problems is testing a package for its ability to recover from errors.**

A controlled test environment usually does not subject the product to the stress encountered during normal use. Therefore, it is difficult to determine how it will perform under prevailing operating conditions. Questions about frequency of errors, ease of recovery, amount of technical assistance required, time to recover, etc. usually cannot be answered until working experience is gained.

11. **It is difficult to assess whether a package is compatible with my existing software.**

This problem was raised as part of the overall difficulty of adequately testing a package. It was suggested that vendors are often not fully aware of which software is compatible with their product.

12. **It is difficult to assess whether a package is compatible with my existing hardware.**

The essence of this problem is knowing the specific configuration required; e.g., often more disk drives or additional main memory are needed in order to run the package. Sophisticated buyers who rely on their own technical experts rather than vendor opinion found this to be less of a problem. This suggests that overall it may become a more significant problem as the distribution of hardware, especially micro-computers, throughout the organization becomes broader.

13. User documentation is not available.

The absence of adequate documentation oriented towards the end user of a package was viewed as a problem regardless of the organization's technical sophistication. This problem is likely to increase as computer usage spreads throughout the organization and greater reliance is placed on direct end user interaction with packages.

14. Systems documentation is not available.

This problem concerns the lack of technical documentation detailing the architecture and functioning of the system. This type of documentation is necessary in order to fine tune the package, modify or customize it, etc. This documentation also needs to be maintained and updated as the system changes. A lack of systems documentation is more serious in large packages than in small ones.

15. Applications documentation is not available.

This concerns an intermediate level of documentation which explains how the system performs particular functions, its limitations and constraints, etc. Documentation of this sort is necessary for effective operation and monitoring of a package.

Problems related to product modification

16. Modification costs are too high.

The price quoted to modify a package to meet the user's specific requirements is often higher than the purchase price for the package. In theory, this imbalance should be expected; the package price is based on the expected sale of a number of copies, while modifications are one of a kind. Nonetheless, it is often viewed as a problem.

17. Modification costs cannot be accurately estimated.

Three factors were thought to contribute to this problem: buyer indecision, legitimate inaccuracy, and vendor 'game playing' (or illegitimate inaccuracy). The first arises when the buyer revises specifications after costs have been estimated, and changes which appear trivial to the buyer are costly to satisfy. Legitimate inaccuracy occurs because modifications are in fact unique, and hence difficult to estimate accurately. Vendor game playing, e.g., underestimating modification costs in their zealously to close the deal, probably contribute least to this problem but are cited the most frequently by buyers.

18. The time required for modifications cannot be accurately estimated.

The factors at work here are the same as those which impact cost estimates. However, timing inaccuracy has a more profound effect on the organization. Various activities are scheduled around the planned installation of new software; old systems may be phased out, client relationships may be reassigned, the departmental framework may be recast, marketing programs may be implemented. The problem is not the length of time per se, but rather the negative impact on contingent activities.

19. Modification of packages is too time consuming for my needs.

This is related to the previous problem, but concerns the actual length of time required. Buyers do not want modifications to 'unreasonably' prolong the installation process; and, it appears that what is reasonable to the vendor is often viewed as unreasonable by the buyer. Since total elapsed time is often a key argument for installing a package, this can become a serious issue. Dissatisfaction with the time required to modify packages seemed most significant when the vendor made the modifications, probably reflecting the difference in priorities between buyers and sellers.

20. Software package vendors are not reliable in either their cost or time estimates for modification.

This statement suggests that the problems with packages lie squarely with the vendor, an issue addressed more fully in the fourth group of problem statements. At its heart it concerns the

vendor's ability to service the product. Since service is often a major portion of what one buys when acquiring a software package, this is in most cases a critical issue.

Problems related to internal organizational factors

21. It is often difficult to reach an internal consensus regarding a particular package due to the differing criteria of programmers, end-users and management.

Technical, end-user and financial criteria are all important considerations in the software acquisition process. They are often, however, incompatible, and must be traded off against one another. Many respondents advocated stronger user involvement in the decision process as a solution to this problem. As users increase their knowledge and technical expertise, they will be in a position to weigh all factors, and will be able to use technical staff as consultants rather than as decision makers. This suggestion may reflect a user bias in our sample, as it ignores DP's legitimate need to influence software decisions in order to assure smooth operations.

22. Our internal procedures cannot be changed to accomodate the structure of a package.

Organizations are composed of a multitude of interacting functions. Changing the way one of those functions is performed cannot be considered in isolation, but only in the context of the other functions with which it is interdependent. In large organizations the difficulties this raises are substantial, and

procedural changes are avoided. These organizations typically have sophisticated technical staffs, and it is easier and more efficient to modify the product. In only one of the organizations interviewed was there support for the reverse position -- the necessity to change internal procedures to accommodate a package -- and this was the smallest and technically least sophisticated of all organizations in the sample.

23. Software evaluation specialists are too difficult to find.

Several interviewees expressed the need for an independent source of software package expertise. These software evaluation specialists could be hired as consultants on a project by project basis, or they could be members of a separate internal consulting staff independent of the DP staff. Independent and knowledgeable consultants would be able to perform this software evaluation function objectively and apolitically.

24. It is difficult to arrange for end-users to test the product before buying.

Though the importance of end-user participation in the selection process cannot be denied, users often prefer to defer to the "expert" opinion of the DP staff. In part, they may feel threatened by the concept of automation; in part, they feel their level of knowledge is too limited to make a good decision. Whatever the reason, they give priority to other activities.

Compounding the problem is a lack of vendor cooperation. A test installation of the software on the buyer's premises is time consuming and costly, and hence, avoided if possible.

25. It is difficult to translate the needs and demands of end-users to package specifications.

This is closely related to the problem discussed above. Users are reluctant to participate in the acquisition process to the extent needed to develop detailed, thorough specifications. It was suggested that users are most willing to participate at the initial stage of defining needs, but want the DP staff to then handle contacts with vendors, etc. Greater user participation at all stages is needed in order to assure that specifications are complete and accurate.

26. Our organization may have traces of a 'not invented here' complex.

Many respondents suggested that systems people were the most guilty of holding this attitude. They rationalize the need to have systems which are developed in-house by stating that you cannot fully understand, and hence cannot maintain, service or modify, a system unless you have developed it from scratch. A more realistic explanation may be that programmers find it more fulfilling to create than to modify someone else's work.

Problems related to vendor relationships

27. Legal negotiations with vendors slow the purchase process.

The legal status of computer software is vague and and not clearly established. Issues of responsibility, obligations, and measurement criteria must be negotiated, often by attorneys who have inadequate knowledge about software.

28. Lack of standard legal parameters make vendor obligations unclear.

This issue is closely related to the previous one. There are few accepted industry-wide legal standards for software. The rights and obligations of both buyers and vendors are uncertain. Many of the stated (or implied) contractual agreements are not clearly enforceable, leaving the courts to be the ultimate authority.

29. A warranty or insurance policy is not usually available.

Warranties are based on the Uniform Commercial Code, but these apply only to the sale of goods. While software is considered a good, the inclusion of services make the scope of warrantry protection under the UCC in these matters unclear (see [8]).

30. The financial stability of most small vendors is questionable and cause for real concern.

The demand for software has produced a flood of entrants in the field, some of whom have produced excellent products. The low cost of entry, however, results in many firms not having

sufficient financial backing or experience to remain in business. Further, it is difficult to get needed financial data to review the stability of newer firms.

31. Vendors are generally slow in updating their products.

In all packages there are areas that could be updated and improved. The vendor, however, generally focuses his energies on selling existing products and developing new ones, not on improving products already sold and installed.

32. There are no accurate or standard assessment devices for evaluating vendors.

The purchase of a software package is the purchase of both a product and a service, and the overall quality of the package is often very dependent on the service received. While there are several publications and organizations which regularly rate the product part of the package, there is no comparable rating of the vendors. Buyers must rely on informal and anecdotal evaluations of the vendors and the service they provide. This is complicated by frequent personnel changes and differences in service provided by the same vendor in different geographic locations.

33. Vendors do not generally service their packages on a timely basis.

This is closely related to the problem of slowness in updating packages. Servicing existing installations is not as profitable as selling new ones, in the short run at least. In addition, the vendor seldom feels the same urgency that the

buyer/user feels to keep the package running; what to the vendor may seem a very fast response to the problem may be an intolerable delay to the user.

34. Hardware manufacturers do not provide enough information on compatible software packages.

Although several hardware manufacturers issue periodic listings of software, they are not usually detailed or comprehensive nor do they rate the packages or include evaluations of the vendors. There are obvious reasons for this, e.g., the manufacturers own proprietary interests, potential anti-trust problems, etc. Nonetheless, this means that one potentially valuable source of information about packages is not as useful as it might be.

35. Software vendors do not understand my industry category.

The requirements in some industries may be so different as to render general software packages inadequate. Large financial institutions were most vocal about this, and cite this as their reason for maintaining large internal staffs. Several respondents noted that they rely heavily on software vendors which they regard as industry specialists.

36. Vendors are not willing to demonstrate their products on my premises, using my hardware.

This is the flip side of the problem with arranging end-user tests prior to purchase. Vendors prefer to do canned sales presentations. Tests on customer equipment at customer sites are

expensive and time consuming. Further, since the circumstances are likely not optimal for the product's performance, such demonstrations do little to help the vendor sell his product.

37. Once a package is in-house (either rented, leased or bought) vendors do not adequately support it.

This again relates to the service provided by the vendors, though this time the problem is phrased more broadly. The frequency with which this problem (or its variants) arose in the interviews suggested that it was one of the more critical concerns of package buyers.

38. Demonstrations are only effective in showing the 'bells and whistles' but do not fully detail the specifics of the package.

The vendor views the demonstration as an opportunity to sell the firm's image as well as the product. He must present to a buying group consisting of representatives from various departments. Some members of the group have carefully studied the package and are looking for detailed information to help assess its merits. Other group members have minimal knowledge of the package or of computer technology in general. The vendor must tailor his pitch to the lowest common denominator. The 'bells and whistles' are usually the strongest sales pitch to this mass audience.

Questionnaire Survey Results

The initial interviews provided a broad list of problems but did not enable us to gauge their relative importance in the package acquisition process. The questionnaire survey addressed this question. Exhibit 3 shows the mean score assigned to each of the 38 problem statements by the 55 questionnaire respondents. Mean scores range from 1.745 (something less than "a minor problem") to 3.836 (slightly less than "an important problem") on a 1 to 5 scale. The median problem had a mean rating of approximately 2.95 (roughly "somewhat of a problem"). An analysis of Exhibit 3 suggests some important patterns in the results.

Insert Exhibit 3 about here

If there is a single theme which runs through all of the problems rated as important it is uncertainty; uncertainty about package capabilities, the time and cost required to install the package, and the ability to maintain it. The problem area considered most important is the difficulty of estimating the time and cost of package modification or customization. Three statements which address this (#18, 17 and 16) are ranked 1, 2 and 3 overall. This problem, however, is not viewed as being necessarily the vendor's fault, as evidenced by the lower ranking (10th) of problem statement #20. A second important issue is the

financial stability of the vendor (item #30), which ranks 4th. Related to the issue of stability is that of the availability of documentation and source code; problem #'s 14, 15, 1 and 13, which rank 5th through 8th, respectively. Another important problem area concerns the ability to specify needs and to test the package to assure that it meets those needs (problem statements #25, 38, 8, 9, 10 and 24, ranked 9th, 11th through 14th, and 16th).

One problem area that surfaced frequently in the interview phase but was rated as only of average importance in the survey phase concerns relationships between buyers and vendors -- the buyers' ability to assess the vendors and the service and support provided by the vendors. Items 28, 32, 33, 31 and 29 are all concerned with this issue and are ranked 17th through 20th and 22nd. Other items relating to vendor relationships were ranked even lower.

A number of items ranked at the bottom of the list focused on the organization's ability to assess the package, its fit with existing hardware and software, etc. (items #2, 3, 34, 36, 11, 23 and 12, ranking 30th through 32nd and 35th through 38th). Apparently, buyers have relatively little trouble assessing the fit of packages to their technical environment and do not need additional specialized expertise in this area.

Insert Exhibit 4 about here

In addition to analyzing the responses of the survey group as a whole, we wanted to see if there were differences in problem perceptions based on the role played in the software acquisition process. Respondents were asked to indicate whether their orientation in software acquisition was primarily technical, financial, or as an end-user. We were able to identify sub-samples which played each of these roles. Those included in the technical group (n=7) described their orientation as either technical or technical and end-user. Those in the financial group (n=6) described their orientation as either financial or financial and end-user. Those in the end-user group (n=13) described their orientation as being solely that of an end-user. Exhibit 4 shows the rank ordering of the 38 problem statements by each of these sub-samples. These sub-groups are small, and represent only about half of the total sample. Thus, any observations we make about them can only be tentative and should not be considered statistically significant. Nonetheless, there are some interesting differences which deserve mention.

All three sub-groups see vendor stability and uncertainty about the time and cost required for modification as important problems. Their relative positions, however, differs from group to group. Vendor stability is the number one problem for the

financial sub-group but drops to number 10 for the end-users. Uncertainty about modification is most important to the end-users, slightly less critical to the financial people, and less still to the technical group. The financial group, however, is quite concerned about the overall cost of modifications.

Beyond this, the profile of the technical sub-sample, generally, is not too different from that of the overall sample. The financial sub-sample, however, does exhibit some consistent differences. As mentioned already, they are more concerned with financial criteria, e.g., vendor financial stability, overall cost level, package efficiency. They also tend to be more concerned with vendor issues, especially the inadequacy of vendor support; e.g., items #31, 33 and 37, which they rank 3rd, 9th and 20th vs. 20th, 19th and 27th in the full sample. They seem less concerned about the availability of technical documentation (items #14 and 15) which they rank as 28th and 29th vs. 5th and 6th in the overall sample. Finally, they rank those items dealing with legal issues much lower than do the other sub-groups or the full sample (e.g., items #28, 29 and 27, ranked 17th, 22nd and 29th overall, were ranked 33rd, 31st and 35th by the finance group).

The end-users, in general, display more concern with user issues and show less ability to independently evaluate vendors and packages. Thus, ability to do end-user tests prior to purchase (item #24), the problems of trading off among factors

(item #4), the availability of packages which reflect industry conditions (item #6), as well as the lack of software evaluation specialists (item #23) and the difficulty of assessing compatibility with hardware (item #12) are all rated higher by end-users than by any other group. One surprise is the relatively low rating they give to the lack of availability of user documentation (item #13). Perhaps users expect to rely on someone within the organization to provide them with documentation and instruction in system use. Thus, they do not view the absence of vendor provided documentation as a problem; someone in their DP shop will provide what documentation is needed.

Discussion and Recommendations

The results of this study suggest that the key issue which needs to be addressed in the application software package acquisition process is uncertainty. Three types of uncertainty are highlighted in the responses to the survey questionnaire:

1. uncertainty about time and cost for package modification;
2. uncertainty about vendor viability, and its implications for package maintenance; and
3. uncertainty about precise user needs and the package's ability to meet those needs.

In this final section we will explore briefly some steps which might be taken by buyers, vendors, and the software industry in general to address these three sources of uncertainty.

Uncertainty about time and cost of modification

Software construction and modification remains largely an art. Further, the more modifications one makes to a system, the more likely it is that unforeseen problems will arise which take time and money to resolve (see [5]). One way to address this problem is to limit the amount of modification made. While we have no direct evidence of this, it is our belief that much modification is done without a careful analysis and evaluation of the alternatives. The costs and risks of changing organizational procedures should be compared to those of modifying the package. In many cases it may prove that simple changes to the organization and the way it does things will be less costly and take less time than trying to modify a software package. This is especially true for smaller organizations which do not have large, highly skilled technical staffs to perform the software modifications and to maintain them. We note that item #22, the difficulty of modifying organizational procedures to fit an existing package, ranked no higher than 24th in any analysis performed on these data.

Reducing the extent of requested modifications is something the buyer/user can do. The vendor can help in this process by being very clear about the ease or difficulty of making various

modifications. Buyers may request certain modifications which to them appear simple but, in fact, are quite complex. The vendor, anxious to make the sale, may not mention the difficulty of that change. This leads to unrealistic expectations on the buyer's part, something which should (and could) be avoided.² There may be several alternative modifications among which the buyer is indifferent; but, unless the vendor makes clear the difficulty of each change, the buyer has an inadequate basis for making his decision. What is needed is open communication between vendor and buyer in order to form a shared perspective. This dialogue must include the end users; it will not suffice to involve only DP, and to allow them to make these decisions without end user input.

A third approach to solving this problem is more technical. Packages can be designed with more built-in flexibility, options which can be specified by the user. The incorporation of report generators, query languages, DBMS, non-procedural languages, etc. into applications packages will enable the end-users to make many of the needed modifications. While this approach cannot obviate the need for all modifications, it should, in combination with the other steps suggested here, go a long way towards reducing the amount of additional programming required to install a package. This, of course, will reduce the uncertainty about the time and cost of modification.

Uncertainty about vendor viability

The concern about the financial stability of small vendors is not a genuine concern about vendor health; rather it is concern about how a package will be updated and maintained should the vendor go out of business. Dealing only with large, well established vendors would be one way to address this problem, but it has a serious disadvantage: much of the interesting software that an organization would like to acquire is produced by small firms with limited capitalizations and perhaps unstable futures. Software buyers will probably need to continue dealing with smaller and less secure vendors for some time to come.

Several steps can be taken to protect buyers in case of a vendor failure. First, routine provisions should be made to make source code available in the event a vendor is no longer able to maintain its products. Perhaps some type of industry association could be formed that would serve as a repository for package source code. This could serve the buyer's needs of assuring that the code was available while at the same time protecting the vendor's proprietary interests. To be most effective, this repository should assure that it receives updates of the code whenever new versions are released. Other, related steps would include having standards for code (e.g., structured programming) to assure that it could be maintained by someone not previously familiar with it, having standards for documentation, and routinely providing that documentation to package purchasers or

licensees. There is a place here for real cooperation among vendors and between vendors and users, which should result in benefits to all parties. The fact that buyers do not perceive vendors to be the source of existing problems suggests that there is a good basis for establishing greater vendor-user cooperation.

Uncertainty about user needs and package capabilities

This problem is perhaps best addressed in two parts. Uncertainty about user needs is not (or certainly should not be) unique to software packages, but is equally a problem in cases of in-house development. The answer in both cases is to have better, more thorough user involvement throughout the selection, development and acquisition processes.

Uncertainty about package capabilities is something which vendors should address. They must recognize that they are no longer (if they ever were) in the business of selling a simple product. Rather, they are selling a combination of product and service. Further, they are dealing with an ever widening audience, incorporating more sophisticated and knowledgeable users as more large organizations turn to packages, and more naive users as mini and micro-computers find their way into more and more organizations. The different parts of this audience need different services, information, etc. More technically sophisticated users (including DP professionals who may buy packages for end-users in their organizations) want considerably more detailed product and performance information than most

vendors have given out in the past. The overview of 'bells and whistles' which may be adequate for the computer-naive end-user will not satisfy the DP professional. Some of the steps which vendors can take to address these problems are: (1) employ industry specialists who not only know the product, but also know the needs of particular industries; (2) generally upgrade the training of sales personnel to make them more like management consultants; (3) price products to include an appropriate level of service; and (4) provide maintenance contracts, telephone hot-lines, seminars, and other devices to enable potential users to get answers to important questions and actual users to get their problems solved.

* * * *

The applications software package industry is expanding rapidly despite the problems discussed here. Solving those problems, however, should not be too difficult and will be worthwhile for vendors and buyers alike. We need to recognize, though, that accomplishing this will require the cooperation of both sides. It is not something which can be left to the vendor alone. The active participation of user organizations at all levels is needed.

References

1. Alloway, R. & J.A. Quillard, "User Managers' Systems Needs," MIS Quarterly, Vol. 7, No. 2 (June, 1983), pp. 27-41.
2. Berkeley, A.R., "Millionaire Machine?" Datamation, Vol. 27, No. 8 (August 25, 1981), pp. 20-36.
3. Blumenthal, M., "Revenues Up 21% in 1980 for Computer Services," Computerworld, Vol. 15, No. 27 (July 6, 1981), pp. 73-75.
4. Blumenthal, M., "Adapso: Revenues Up Average 23% in All Areas of Software, Services," Computerworld, Vol. 16, No. 26 (June 28, 1982), p. 67.
5. Frank, W.L., "Commercial Software," Computerworld, Vol. 13, No. 52 (December 24, 1979), pp. 17-22.
6. Ginzberg, M.J., "Early Diagnosis of MIS Implementation Failure: Promising Results and Unanswered Questions," Management Science, Vol. 27, No. 4 (April 1981), pp. 459-478.
7. "Jobs for Programmers Begin to Disappear," Business Week, No. 2752 (August 16, 1982), p. 29.
8. Lavey, W.G., Esq., "Some Legal Aspects of Acquiring an Applications Software Package," Harvard Library of Computer Graphics, Vol. 12 (1981).
9. Runyan, L., "Applications Development: Software Still a Soft Spot," Datamation, Vol. 27, No. 3 (March 1981), pp. 165-166.
10. Snyders, J., "What's Coming Down the Pike in Software," Computer Decisions, Vol. 13, No. 12 (December 1981), p. 122.
11. "The Switch to Specialized Programs," Business Week, No. 2735 (April 19, 1982), pp. 92A-92D.
12. "Top 100 U.S. Companies in the DP Industry," Datamation, Vol. 26, No. 7 (July 1980), pp. 87-94.

FOOTNOTES

1. These data are taken from the 1980 Datamation survey of the U.S. computer industry [12].
2. See [6] for a discussion of the importance of realistic user expectations to successful implementation.

PROFILE OF ORGANIZATIONS INCLUDED IN
IN-DEPTH INTERVIEWS

Advertising

2 agencies ranked in top 10 by billings

Banking

2 major transnational banks, both serving
corporate and retail customers

Publishing

1 medium sized financial magazine
1 medium sized computer service publisher

Education

1 large graduate school

Energy

1 Fortune 500 energy corporation

Manufacturing

1 large industrial and consumer product
manufacturer

Financial Services

2 major financial services companies

Exhibit I

SURVEY RESPONDENTS BY FIRM AND INDUSTRY

	<u># of respondents</u>
<u>Advertising</u>	
Agency 1	2
Agency 2	2
Agency 3	1
<u>Banking</u>	
U.S. Bank 1	1
U.S. Bank 2	2
U.S. Bank 3	4
U.S. Bank 4	3
Foreign Bank	1
<u>Financial Services</u>	
Mortgage Service	1
Diversified Financial 1	4
Diversified Financial 2	1
Investment Banker	4
Depository	2
<u>Education</u>	
Professional School	1
Large City Board of Education	1
<u>Manufacturing</u>	
Misc. Manufacturing 1	1
Misc. Manufacturing 2	1
<u>Insurance</u>	
Large Insurer 1	3
Large Insurer 2	3
Medium Insurer	1
<u>Publishing</u>	
Record Producer	2
Financial Magazine	1
DP Industry Publisher	1

(continued on next page)

SURVEY RESPONDENTS BY FIRM AND INDUSTRY (continued)

	<u># of respondents</u>
<u>Energy</u>	
Major Oil 1	1
Major Oil 2	1
Major Oil 3	1
Major Oil 4	1
Driller	1
<u>Public Utility</u>	
Electric Utility	2
<u>Communication</u>	
Communication Services	1
<u>Information Processing</u>	
Specialized Systems Producer	1
Mainframe Producer	3

Exhibit II

MEAN PROBLEM SCORE
ALL RESPONDENTS (N=55)

<u>Item #</u>	<u>Problem Statement</u>	<u>Rank</u>	<u>Mean Rating</u>
18.	The time required for modifications cannot be accurately estimated.	1	3.836
17.	Modification costs cannot be accurately estimated.	2	3.818
16.	Modification costs are too high.	3	3.764
30.	The financial stability of most small vendors is questionable and cause for real concern.	4	3.618
14.	Systems documentation is not available.	5	3.545
15.	Applications documentation is not available.	6	3.545
1.	Source code is not usually available.	7	3.519
13.	User documentation is not available.	8	3.418
25.	It is difficult to translate the needs and demands of end-users to package specifications.	9	3.382
20.	Software package vendors are not reliable in either their cost or time estimates for modifications.	10	3.327
38.	Demonstrations are only effective in showing the 'bells and whistles' but do not fully detail the specifics of the package.	11	3.278
8.	One of the most difficult problems is testing a package for functional specifications.	12	3.204
9.	One of the most difficult problems is testing a package for efficiency.	13	3.167
10.	One of the most difficult problems is testing a package for its ability to recover from errors.	14	3.167

Exhibit III (continued)

<u>Item #</u>	<u>Problem Statement</u>	<u>Rank</u>	<u>Mean Rating</u>
19.	Modification of packages are too time consuming for my needs.	15	3.167
24.	It is difficult to arrange for end-users to test the product before buying.	16	3.091
28.	Lack of standard legal parameters make vendor obligations unclear.	17	3.073
32.	There are no accurate or standard assessment devices for evaluating vendors.	18	3.000
33.	Vendors do not generally service their packages on a timely basis.	19	2.963
31.	Vendors are generally slow in updating their products.	20	2.944
21.	It is often difficult to reach an internal consensus regarding a particular package due to the differing criteria of programmers, end-users and management.	21	2.927
29.	A warranty or insurance policy is not usually available.	22	2.868
7.	My needs are too unique to be adequately represented in available packages.	23	2.852
22.	Our internal procedures cannot be changed to accommodate the structure of a package.	24	2.836
5.	There is too great a dependence on the vendor's evaluation of packages and not enough third party or objective evaluation.	25	2.796
4.	There is an irreducible trade off among the factors of ease-of-use, flexibility and overhead.	26	2.769
37.	Once a package is in-house (either rented, leased or bought) vendors do not adequately support it.	27	2.722

Exhibit III (continued)

<u>Item #</u>	<u>Problem Statement</u>	<u>Rank</u>	<u>Mean Rating</u>
6.	Available packages do not adequately reflect my industry.	28	2.685
27.	Legal negotiations with vendors slow the purchase process.	29	2.655
2.	Similar products are difficult to compare due to a lack of industry standard.	30	2.574
3.	The trade-offs present in either a 'pure' buy or a buy with modifications decision make the choice difficult.	31	2.556
34.	Hardware manufacturers do not provide enough information on compatible software packages.	32	2.444
35.	Software vendors do not understand my industry category.	33	2.407
26.	Our organization may have traces of a 'not invented here' complex.	34	2.345
36.	Vendors are not willing to demonstrate their products on my premises, using my equipment.	35	2.259
11.	It is difficult to assess whether a package is compatible with my existing software.	36	2.185
23.	Software evaluation specialists are too difficult to find.	37	2.145
12.	It is difficult to assess whether a package is compatible with my existing hardware.	38	1.745

RANK ORDER BY MEAN

<u>ITEM</u> <u>#</u>	<u>RANK</u>			
	<u>OVERALL</u>	<u>TECHNICAL</u>	<u>FINANCIAL</u>	<u>END-USER</u>
18. The time required for modifications cannot be accurately estimated.	1	6	4	2
17. Modification costs cannot be accurately estimated.	2	7	5	1
16. Modification costs are too high.	3	8	2	9
30. The financial stability of most small vendors is questionable and cause for real concern.	4	5	1	10
14. Systems documentation is not available.	5	1	28	4
15. Applications documentation is not available.	6	2	29	5
1. Source code is not usually available.	7	3	6	6
13. User documentation is not available.	8	4	10	18
25. It is difficult to translate the needs and demands of end-users to package specifications.	9	13	13	13
20. Software package vendors are not reliable in either their cost or time estimates for modifications.	10	11	7	7
38. Demonstrations are only effective in showing the 'bells and whistles' but do not fully detail the specifics of the package.	11	9	11	22
8. One of the most difficult problems is testing a package for functional specifications.	12	17	14	14
9. One of the most difficult problems is testing a package for efficiency.	13	24	8	8
10. One of the most difficult problems is testing a package for its ability to recover from errors.	14	10	24	12

Exhibit IV

EXHIBIT IV (continued)

ITEM #	RANK			
	OVERALL	TECHNICAL	FINANCIAL	END-USER
19. Modification of packages are too time consuming for my needs.	15	12	16	19
24. It is difficult to arrange for end-users to test the product before buying.	16	14	12	3
28. Lack of standard legal parameters make vendor obligations unclear.	17	18	33	11
32. There are no accurate or standard assessment devices for evaluating vendors.	18	16	15	30
33. Vendors do not generally service their packages on a timely basis.	19	15	9	15
31. Vendors are generally slow in updating their products.	20	22	3	34
21. It is often difficult to reach an internal consensus regarding a particular package due to the differing criteria of programmers, end-users and management.	21	19	17	20
29. A warranty or insurance policy is not usually available.	22	20	31	23
7. My needs are too unique to be adequately represented in available packages.	23	27	18	35
22. Our internal procedures cannot be changed to accommodate the structure of a package.	24	25	30	24
5. There is too great a dependence on the vendor's evaluation of packages and not enough third party or objective evaluation.	25	23	32	25
4. There is an irreducible trade off among the factors of ease-of use, flexibility and overhead.	26	31	19	16
37. Once a package is in-house (either rented, leased or bought) vendors do not adequately support it.	27	26	20	36

EXHIBIT IV (continued)

ITEM #	RANK			
	OVERALL	TECHNICAL	FINANCIAL	END-USER
6. Available packages do not adequately reflect my industry.	28	30	21	17
27. Legal negotiations with vendors slow the purchase process.	29	21	35	38
2. Similar products are difficult to compare due to a lack of industry standard.	30	33	36	26
3. The trade-offs present in either a 'pure' buy or a buy with modifications decision make the choice difficult.	31	28	22	31
34. Hardware manufacturers do not provide enough information on compatible software packages.	32	36	25	27
35. Software vendors do not understand my industry category.	33	35	34	28
26. Our organization may have traces of a 'not invented here' complex.	34	29	23	33
36. Vendors are not willing to demonstrate their products on my premises, using my equipment.	35	32	26	32
11. It is difficult to assess whether a package is compatible with my existing software.	36	37	27	37
23. Software evaluation specialists are too difficult to find.	37	34	37	21
12. It is difficult to assess whether a package is compatible with my existing hardware.	38	38	38	29