REPORT BY THE U.S.

General Accounting Office

Federal Agencies Could Save Time And Money With Better Computer Software Alternatives

Developing new software for each proposed application is expensive and time consuming. However, cheaper and quicker alternatives, which include reuse of existing software and new development techniques, are available, are more reliable, and can save money.

Federal agencies have generally lagged behind the private sector in using the new alternatives. GAO recommends several specific actions to stimulate the use of these alternatives, improve operations, and save time and money.

GAO/AFMD-83-29
MAY 20, 1983

Request for copies of GAO reports should be sent to:

U.S. General Accounting Office
Document Handling and Information
Services Facility
P.O. Box 6015
Gaithersburg, Md. 20760

Telephone (202) 275-6241

The first five copies of individual reports are free of charge. Additional copies of bound audit reports are $3.25 each. Additional copies of unbound report (i.e., letter reports) and most other publications are $1.00 each. There will be a 25% discount on all orders for 100 or more copies mailed to a single address. Sales orders must be prepaid on a cash, check, or money order basis. Check should be made out to the "Superintendent of Documents".

UNITED STATES GENERAL ACCOUNTING OFFICE
WASHINGTON, D.C. 20548

ACCOUNTING AND FINANCIAL
MANAGEMENT DIVISION

B-210595

The Honorable David A. Stockman
Director, Office of Management
and Budget

The Honorable Gerald P. Carmen
Administrator of General Services

The Federal Government is the world's largest user of automatic data processing (ADP) resources. The cost of these resources runs into billions of dollars annually. An ever increasing share of data processing costs is spent for acquisition of application software.

Application software consists of the computer programs, files, and documentation which automate the tasks of end users; for example, payroll, social security, and air traffic control. Development of such software is the primary limiting factor in what can now be accomplished with computers. Software for complex applications can take years to develop. In some cases, hundreds of staff-years are spent to develop, test, and install software for large applications. As computer hardware costs come down and the feasibility of using computers goes up, user demand for automation increases. This increased demand and the cost of development have caused backlogs of unfilled requests for new applications and for major system changes.

OBJECTIVES, SCOPE, AND METHODOLOGY

We undertook this review to identify (1) problems Federal agencies have in satisfying their application software needs, (2) options available to agencies to acquire application software, (3) whether agencies were taking advantage of the most beneficial options, and (4) recommendations that would help satisfy Federal software needs faster and more economically.

From earlier GAO reports, we reviewed and categorized historical problems in the acquisition of application software. We discussed the subject with software experts, reviewed current literature, and identified and evaluated the options available to satisfy application software needs from trade journals, published market surveys, and contact with various software sharing organizations. We visited 15 Federal data processing installations to see where they had gotten their application software and get some idea of the

extent to which alternatives to new development were being used. We reviewed agency procedures for the software selection process to determine the adequacy of the guidance given to data processing managers on how to select and evaluate alternatives to meet software needs. We also examined applicable central agency guidelines on software acquisition.

Besides visiting the 15 installations, we obtained information from 284 other Federal data processing installations through a questionnaire. Data on commercial software and the Federal Government's participation in that area were obtained through questionnaires completed by 146 software vendors. We used these questionnaires to see if conditions found at the 15 sites visited were widespread.

We performed our review in accordance with generally accepted government audit standards.

ROLES OF CONCERNED AGENCIES

The basic law governing Federal ADP management is the Brooks Act, Public Law 89-306. Under this act, the General Services Administration (GSA) is responsible for coordinating the procurement and maintenance of Federal ADP resources.

GSA receives technical advice from the Secretary of Commerce, primarily through the National Bureau of Standards (NBS). Both of these agencies get fiscal and policy guidance from the Office of Management and Budget (OMB). NBS is responsible for providing scientific and technological advisory services to Federal agencies and for developing Federal Information Processing Standards. The Paperwork Reduction Act (Public Law 96-511) further outlines OMB roles. It says, among other things, that the Director of OMB:

--Shall be responsible for initiating and reviewing proposals to improve ADP and telecommunications practices.

--May designate a central collection agency(ies) to obtain information for two or more agencies.

--Will identify initiatives to improve productivity in Federal operations using information processing technology.

--Will identify duplication and develop a schedule and methods for eliminating duplication.

In addition, each Federal agency has certain responsibilities for managing its own ADP resources. Circular A-71, published in March 1965 by the Bureau of the Budget (now OMB), states that the heads of all executive departments and establishments are responsible for the administration and management of their automatic data processing activities. Circular No. A-121, published in September 1980 by OMB, says that

> "Agencies shall establish cost accounting procedures which are consistent with the Federal Government Accounting Pamphlet Number 4, Guidelines for Accounting for Automatic Data Processing, United States General Accounting Office 1978, for the operation of data processing facilities whose estimated full costs exceed $3 million per year."

The Paperwork Reduction Act says that each Federal agency shall systematically take inventory of its major information systems and ensure that they do not overlap or duplicate the systems of other agencies.

GSA established the Office of Software Development, which now operates the Federal Software Exchange Center and the Federal Software Testing Center. Elsewhere in GSA, the Office of Information Resources Management operates the ADP schedule contracts program, which includes commercial software products.

In our role of aiding the Congress, we are concerned with the management of Federal ADP and with computer software as an expensive part of Federal ADP. Our past reports to the Congress have recommended improvements in ADP management, both Government-wide and at specific agencies.

ALTERNATIVE WAYS EXIST TO MEET SOFTWARE NEEDS

A number of alternative methods can reduce the costs and the delays associated with custom development of new software. The alternatives include purchasing readymade software and sharing software from other organizations. Also, if software must be developed, new methods can be used which can greatly reduce the labor otherwise needed.

The ways now available to satisfy needs for application software include:

- --Making new software through traditional software development. Too often, this method costs more and takes longer than expected and may needlessly duplicate earlier work.

- --Making new software using labor saving aids, called generators and problem oriented packages. This method can greatly reduce labor and time.

- --Using existing software in the form of packages available for sale from vendors.

- --Using existing software by sharing with other Federal agencies, States, and/or user groups.

--Using existing software that now automates the desired application unsatisfactorily, by modifying and enhancing it so that it will be satisfactory.

Once a need for software to automate an application is recognized, the most cost-effective method of satisfying the need should be selected. Federal agencies should carefully explore alternative methods--in addition to new development--of satisfying software needs both to be good managers and to follow guidance from central agencies.

No overall process exists to ensure that Federal agencies consider alternative methods of satisfying software needs. In the past, this was not too important because the only way to get software was to develop it new. Now that there are other ways, however, Federal managers should seek the most cost-effective solutions.

FEDERAL AGENCIES GENERALLY FAIL TO EXPLORE OR USE COST-EFFECTIVE ALTERNATIVES

Federal agencies are making little effort to identify and use today's alternatives to custom development to satisfy their application software needs. Over 98 percent of the application software inventories at 15 data processing installations we visited had been custom developed. Also, Federal respondents to our questionnaire reported that over 95 percent of their software inventories had been custom developed. As indicated, developing large software applications is a long and costly process. Development costs are difficult to estimate accurately at the outset, and often the final costs greatly exceed the original estimates. Long development cycles contribute to the backlog of requests for new applications and changes to existing systems. Backlogs of new application requests awaiting software were identified at most of the 15 data processing sites visited, and 160 of 284 Federal questionnaire respondents indicated they had a backlog of new application requests at their installations.

The 15 sites had acquired only about 1 percent of their application software inventory "off the shelf." Federal questionnaire respondents indicated that about the same percentages held true for their sites. Respondents to our questionnaire to software vendors reported that only about 2 percent of their sales were made to the Federal Government, reinforcing the data from the Federal respondents. Federal agencies have also made little use of the application software available from GSA's Federal Software Exchange Center.

The above evidence indicates that Federal data processing sites overwhelmingly use new development as their preferred way to satisfy software needs. This new development is done by employees and by contractors and is done even for applications that all Federal agencies have in common. Payroll is an example of a common application. At least 78 different Federal civilian payrolls have

been identified--for the most part custom built, unique systems. Private sector firms often use packaged payroll software, and an industry reference publication showed 87 firms that offer general payroll software.

THE PRIVATE SECTOR USES AVAILABLE ALTERNATIVES SUCCESSFULLY

Since developing new software costs so much and takes so long, the reuse of suitable existing software at more than one site can be cheaper and quicker. In the private sector, data processing managers increasingly recognize the many potential benefits of reusing suitable existing application software--often called "packaged software." Many computer installations, both private and Federal, have common tasks--such as payroll--and existing software developed elsewhere, either by agencies or by private vendors, is now available to automate those tasks.

Building new software will continue to get more expensive, while reuse of existing software is relatively cheap now and will get cheaper and cheaper. Some experts cite reuse as the only real solution to today's large-scale software needs. Their opinion is that software productivity cannot be improved enough by better development methods or new programming languages, but only through greater use of existing software. This perception has caused private use of packaged software to grow enough to increase the software industry about 30 percent per year.

For years, some experts have said that, once written, programs should be reused at several sites. The obvious advantage of this is to spread the development cost over more user sites and reduce maintenance costs. Nevertheless, most Federal software today is developed without any structured effort to find existing, suitable software that might do the same job.

Federal agencies lack specific guidelines for selection of the method of satisfying software needs and documenting the alternatives considered. This lack of guidance is accompanied by poor accounting for software costs: in-house development costs tend to be understated. Application software developed by an outside source (a vendor or another agency) is distrusted by Federal agencies. If offered by a vendor, it may be perceived as not meeting Federal needs. If developed by another Federal agency, it typically is shared in an "as is" condition with no mechanism whereby the originator can be paid for helping the recipient with installation or modification. Besides these objections--which are sometimes valid but which were formerly voiced in the private sector also--some agency data processing staff have a negative, "not invented here" attitude.

SUCCESSFUL USE OF ALTERNATIVES BY SOME FEDERAL AGENCIES INDICATES THEY CAN BE USED IN GOVERNMENT

Despite the general lag in Federal use of packaged software and other alternatives, a few Federal agencies have initiated cost-effective solutions to their software needs:

--The Bureau of Reclamation's personnel and payroll system is used by both the Department of Energy and the Department of Education, thereby avoiding redundant new development.

--ACTION hired a contractor to modify an existing accounting system for its use, saving three-fourths of the development cost and three-fourths of the development time estimated by other contractors who proposed to build new accounting systems.

--Fourteen Federal agencies have used software available from the National Association of State Information Systems.

Federal agencies have many applications in common, including personnel, payroll, case control, and inventory control. Some of these functions--for example, paying General Schedule employees--are required by law to be the same in all agencies. These common requirements, coupled with modern, high-level languages that enable the writing of applications so they can be used on more than one brand of computer, demonstrate a huge unrealized potential for sharing systems in the Government.

CONCLUSIONS

Available alternatives to traditional software development can significantly reduce the cost and time needed to satisfy application software needs. These include use of already-existing software (either buying or sharing), use of more convenient ways of developing new software, and use of prefabricated software components.

These alternatives are being used very successfully in the private sector, but Federal agencies overwhelmingly use newly written software for each application. This new software is written either by in-house employees or by contractors. While there is some use of existing application software (bought and shared), this use lags behind the use in the private sector. Since individual agencies and the Federal Software Exchange Center operate sharing activities, there may be some duplication of effort.

The reasons for limited Federal use of more cost-effective ways to satisfy software needs include:

--Specific software selection guidelines are lacking.

--Visibility of software development costs is poor.

--Agency officials have negative attitudes toward seeking alternatives.

--Some application software developed in one organization many need adaptation to be usable by another but there is now no mechanism for the developing agency to be paid for helping the recipient with installation or modification.

--Information on available alternatives is lacking.

The high cost of Federal software demands that serious consideration be given to better ways of satisfying software needs. The viability of using application software that already exists in the Federal sector needs to be better demonstrated to remove agencies' objections to it.

RECOMMENDATIONS

To encourage the use of more cost effective methods of satisfying software needs, we recommend that:

--The Director, OMB direct (1) agencies that develop applications and (2) agencies that operate software sharing activities, to make their software, documentation, and directories available to the Federal Software Exchange Center operated by GSA's Office of Software Development.

--The Director, OMB analyze the possibility of combining some of the other agencies' software sharing efforts with the Exchange Center's efforts to reduce duplication.

--The Administrator of General Services direct that:

- The schedule contracts branch of GSA's Office of Information Resources Management require vendors to complete a standard software summary (SF-185) on each software product for which they negotiate a contract and forward the summaries to the Federal Software Exchange Center for inclusion in the catalog section that deals with vendor software. This action will give Federal agencies better information about available software and provide free advertising to the vendors.

- The Office of Software Development demonstrate the concept of Federal use of vendor-developed proprietary application software by selecting from one to three vendor-developed application software packages and modifying them for general use by Federal agencies as pilot projects. Payroll, inventory, and case control might be suitable candidates since they are common to virtually all agencies.

- The Federal Software Exchange Center demonstrate the concept of deliberate reuse of federally owned application

software by acquiring, enhancing, and advertising for general use at least one commonly used large-scale application, such as a Federal personnel system, as a pilot project. The enhancement, and the later maintenance of the enhanced application, could be done by a contractor.

--Heads of Federal agencies install formal software selection procedures on how to identify, evaluate, and select ways of meeting software needs, including vendor packages and shared software as well as custom development, and require that the selection process be documented. Our provisional checklist (appendix 1) offers guidance.

AGENCY COMMENTS

We asked for comments from the Office of Management and Budget, the General Services Administration, and the Department of Commerce. They all responded, and their comments are included verbatim as appendix II.

OMB agreed with the general approach recommended in the report, pointed out possible problems with software sharing, and discussed other activities it has underway, including central service centers and shifting of decision responsibility. OMB generally agreed with our recommendations but did not indicate what its response will be.

GSA agreed that alternatives to custom software development are essential, supported the recommendations to OMB, and concurred with the recommendations to heads of Federal agencies. It also agreed to better inform Federal agencies about available vendors. Concerning the recommended pilot projects, GSA proposed alternative actions. While we agree that GSA's alternatives are constructive, we still believe that pilot projects of the type we recommend are needed.

The National Bureau of Standards of the Department of Commerce stated that the report is timely and addresses a very important set of issues associated with software development and use. The Bureau also stated that the report is consistent with its current efforts to develop guidelines for Federal agencies on the specification, evaluation, selection, and testing of applications software packages.

- - - - -

As you know, 31 U.S.C. 720 requires the head of a Federal agency to submit a written statement on actions taken on our recommendations to the Senate Committee on Governmental Affairs and

the House Committee on Government Operations not later than 60 days after the date of the report, and to the House and Senate Committees on Appropriations with the agency's first request for appropriations made more than 60 days after the date of the report.

W. D. Campbell
Acting Director

PROVISIONAL CHECKLIST FOR SELECTING APPLICATION SOFTWARE ALTERNATIVES

This checklist provides suggested procedures that GAO feels agency management should follow in the software selection process. The checklist is divided into three phases--identifying alternatives, evaluating alternatives, and selecting alternatives. It also includes a list of relevant documents.

While we realize certain constraints may limit available alternatives in given cases, each alternative should be fully evaluated to see whether it is usable and, if so, whether it is the best choice.

I. IDENTIFYING ALTERNATIVES

The alternatives now available to satisfy software needs may be divided as follows:

--Make new software with traditional methods:

1. With in-house staff.
2. With a contractor.

--Make new software with labor-saving methods:

1. Convenient-programming aids.
2. Problem oriented packages.
3. Prefabricated components or other existing code for parts of the new system, with the rest newly developed.

--Use existing software:

1. Packages acquired from vendors.
2. Shared software, from Federal or other sources, which is available for copying and conversion costs--less than new development costs.
3. Enhancements of the organization's existing software.

The size of the desired task and the amount and kind of resources available for it can have significant bearing on whether a given alternative should be considered and on how effective it will be if used. Each alternative has variable factors that should be considered. Also, the alternatives may be used in a variety of combinations. Some of these factors and combinations are listed in the discussion of each option. Requirements must be determined before the proper alternative can be identified and selected. At a minimum, these requirements must be stated in functional terms, such as the user task to be done, the required interfaces with other applications, and the processing environment. The more detailed the statement of requirements is, the more accurately the alternatives may be identified and selected. The following criteria may be used to determine case-by-case which alternatives apply.

A. Make new software with traditional methods[1] (if new development is proven necessary):

1. In-house

--Does the in-house staff have the necessary skill (technical and managerial)?

--Can the in-house staff be spared from other projects?

--Is the necessary machine time available for the development effort?

--Can the in-house staff develop the software in the necessary time frame?

2. Contractor

--Consider use of custom software contractor if availability, cost, time, and contractor expertise--and in-house management skill--make this option feasible. GSA published guidance on contracting for custom software development in GSA Bulletin FPMR F-131 ADP and Telecommunications, May 19, 1981.

B. Make new software with labor-saving methods (if new development is proven necessary)

--Is a labor-saving method available for the computer being used?

--What are the conversion ("lock-in") implications of using a labor-saving package? For example, is it available on more than one brand of computer?

--Must staff be trained especially to use some labor-saving package or method?

C. Use existing software (bought, shared, or enhanced)

--Is there software available? Several commercial listings of vendors and software applications are available to aid in locating existing packages.

--Does the software meet the user requirement as is? With modification?

---

[1]The costs of such new software development should be accounted for as set forth in our "Guidelines for Accounting for Automatic Data Processing Costs" (Federal Government Accounting Pamphlet No. 4, 1978) and called for in OMB Circular No. A-121 (Sept. 1980).

--Can user requirements be modified so that the existing software can be used as is, to avoid the cost of modifying the software?

--If the organization's existing software is to be enhanced, are there tools available to aid the process, and how great is the enhancement effort?

--Does the software now run on the same brand of computer that the organization has?

II. EVALUATING ALTERNATIVES

Once all alternatives have been identified, they should be evaluated. Where more than one alternative is viable, the selection of the alternative should be based on some form of cost-benefit analysis. Some of the cost factors to be considered for each alternative are discussed below.

A. MAKE NEW SOFTWARE WITH TRADITIONAL METHODS

1. In-house staff development

--Must additional staff be hired for the project? If so, what will the extra staff do after the project is completed?

--What opportunity costs will be incurred for lengthy development cycles? (Opportunity costs include forgone benefits associated with the new application due to delay and with the other work the development staff could not do.)

--What risks are involved in the in-house development of new software in terms of:

- Cost overruns?
- Calendar overruns?
- Unreliable software?

--The cost to develop the software in-house should be estimated both for cost-benefit analysis, to decide whether to do the automation at all, and for comparison to the cost of other software alternatives.

Estimating costs for in-house software development

The costs for contracting out software development and for obtaining software packages usually are more readily ascertainable than the costs for in-house development efforts. Nevertheless, an agency needs to be able to realistically estimate the cost of in-house development in order to adequately compare it to other alternatives.

Several Government and private documents address the issue of estimating in-house software development costs. The documents show that an organization must first determine its resource costs (personnel, ADP equipment, and so forth) and the factors affecting these costs in order to establish a cost data base. It must then use a sound technique to estimate the overall development costs. An overview of the major resource costs and the factors that can affect them, and of techniques available to estimate total development cost appears below.

Resource costs and factors affecting them

The three major resources that constitute the cost of software development are:

--People. The number of staff-months required for software development is the primary cost element in estimating the total software development cost. The following two factors affecting staff-months need to be considered when estimating this element.

- The number of people assigned. The documents show that more people assigned to a development project can yield lower individual productivity and thus higher costs. Individual productivity is lowered because a larger group must spend more time communicating.

- Their activities. The documents also show that, generally, 40 percent of development effort is spent on analysis and design, 20 percent on coding, and 40 percent on testing. These percentages can be used in adjusting cost data when the new system is not analogous to an agency's existing systems. For example, if an agency has productivity data for coding only, then total staff costs can be extrapolated by using the estimated ratio of coding that will be required for the new system. Methodology also exists for more detailed cost estimation, such as programming time. Some installations have arrived at a cost per line of code based on experience. Others have cost-estimating formulas which incorporate different variables. These variables include size, complexity, number of input/ output devices controlled by the program, programming language, and programmer skill. Values are assigned to these variables to estimate the cost.

--Computer costs. One method of estimating computer costs is to relate the number of hours of computer time to the activity and the number of people assigned at a given time. For example, one study estimates it requires 3 hours during coding, and 15 to 20 hours during testing. These data can be adjusted depending on the type of computer used.

--Elapsed time. Elapsed time is the length of time from start to finish to complete the development project. Although elapsed time is not a directly chargeable cost in itself, for three important reasons it must be considered in estimating total development costs.

- When the automation is delayed, opportunity costs can be significant. Further, delays increase the risk that user requirements will change and that the software will be partly obsolete when delivered.

- Some estimated costs (such as computer costs) increase because of time-dependent factors, such as inflation rates.

- The amount of elapsed time can also affect personnel costs. One analysis shows that too little as well as too much time allocated for a project can increase personnel costs. Too little time means a larger number of people must be assigned to the project. (The impact of assigning too many people was discussed above.) Too much time can cause increased costs due to personnel turnover which results in retraining costs and lower productivity.

Other costs and factors

Other costs and factors can also significantly increase the overall cost of in-house software development efforts. Examples are:

--Preparation of documentation and user manuals.

--Costs for personnel assigned to the development project, including programmers, analysts, support, management, and user representatives.

--Complexity or difficulty of the system or use by the system of a machine oriented language instead of a higher order language.

--Type of system. (It costs more to develop a real-time system than a batch oriented system.)

--Size of system. Analyses show that it costs more per instruction to develop large systems (100,000 instructions) than small systems (5,000 instructions).

--Support and management personnel costs.

Techniques for estimating in-house development costs

In estimating its resource cost rates, an agency has established a cost data base for estimating in-house development costs.

The agency can then use the following techniques to estimate its total in-house development costs.

--Analogy. The cost to develop an existing similar system is used to estimate the cost of a new system. This technique works best when the systems are similar and development methods and resources are the same. This technique does not provide realistic estimates for unique systems.

--Decomposition. This technique involves dividing a system into components down to a level where it is possible to estimate each component's cost with reasonable accuracy. One variation of this technique identifies new system components down to the level of subroutines which closely resemble components of existing systems for which costs are known. By analogy, the cost of the new system can then be estimated. This technique works well for unique systems, but does not lend itself to rapid estimations. It also is not useful for performing preliminary estimates where initial design work has not been performed.

--Parametric models. Analyses of historical development cost data identify cost variables, that is, factors affecting resource costs, and quantify their relationship to cost through equations. This technique lends itself to rapid estimations and can be used by personnel who are inexperienced in software development. However, it cannot be readily used unless the agency has a well-established cost data base.

--Combination of above techniques. To test the validity of an initial estimate, two or more estimates may be made using different techniques. That is, when the analogy technique was used to make the initial estimate, independent estimates can be made using decomposition and parametric models. Differences can then be analyzed to arrive at a consistent estimate.

2. Contractor development

--Agencies can obtain a realistic cost proposal from the contractor by taking the following steps:

1. Describe the needed software as completely as possible in the Request for Proposals so that the contractor can understand and address the full scope of work in the proposal.

2. Describe the software so that people not familiar with agency operations can understand the need.

3. Include all details from the system development steps completed by the agency to date.

4. In the areas where detailed specifications have not been developed, clearly state the functional requirements the software must satisfy.

5. Give all known constraints and parameters the vendor must work with in developing the software.

6. Avoid the use of agency jargon, which might be unclear to outsiders.

7. Determine whether the contractor's accounting system is adeqate for generating valid cost estimates.

8. Compare the cost of past development efforts to current proposed costs.

9. Exercise caution on bids that are much higher or lower than the average of bids received.

--The contractor's proposal should then be evaluated to see whether:

1. The contractor has knowledge about the agency's mission.

2. The contractor is relying on state-of-the-art or unproven methodology as opposed to proven technology.

3. The overall design is sound and feasible.

4. The contractor uses software performance measurement tools and techniques as well as software optimization tools to ensure the most efficient development possible.

5. The contractor's organization reflects adequate management overall.

6. The staff responsible for the proposal will also work on the development.

7. The key personnel will remain on the project from start to finish.

8. The contractor's quality assurance measurement is compatible with the agency acceptance criteria for the final project.

B. DEVELOP NEW SOFTWARE WITH LABOR-SAVING METHODS

--For short-term applications, can generative techniques cut development costs? Under generative techniques, we consider convenient-programming aids and problem oriented packages.

Convenient-programming aids are software translators that use shorthand coding to reduce the labor of programming on many classes of problems. (Thus, they are more general than problem oriented packages.) They may translate the shorthand statements written by the programmer/analyst directly into machine oriented language (as do examples such as EASYTRIEVE and DYL260), or they may translate it into a standard programming language which can then be submitted to its appropriate standard compiler. Examples of the latter case are SCORE and PROMACS, which translate their shorthand inputs into COBOL programs.

Convenient-programming aids allow programming shops to turn around unplanned user requests much more quickly and at lower labor cost. They should be used primarily for short-lived applications, especially if they do not generate a standard language. Convenient-programming aids can also allow users to code some of their own applications because they typically require much less programming detail than traditional, procedural high level languages. Some packages do generate a standard language, and these could be used for longer-lived applications because the standard language they generate could be converted to a different brand of computer if necessary.

Problem oriented packages are software packages that provide quick automation for a particular class of problem. They typically enable the using analyst to automate the problem with brief statements similar to the natural language of the problem. The software then invokes stored logic which does the processing requested by the analyst's brief statements. Examples include the Statistical Package for the Social Sciences (SPSS) and the Statistical Analysis System. Problem oriented packages allow solutions to specific problems to be coded far more quickly at far less programmer labor cost than would be required if the automation were done with custom programming in regular procedural languages. Many such problems require only short-lived automation.

--When the answers are obtained, the code is thrown away. The cost of traditional custom programming is not justified in many such situations, and problem oriented packages allow it to be avoided.[2] These packages also allow more analysts to automate their own analyses without waiting for programming. They should be used primarily for short-lived applications--not for those that might last long enough to be converted--because there is not yet any Federal standard for problem oriented packages to provide portability across different brands of computers. (However, some of the packages offer different versions for different brands of computers, for example, the SPSS.)

---

[2]Indeed, we used the SPSS to analyze the questionnaires for this project.

C. USE EXISTING SOFTWARE

1. Using software acquired from vendors

Data to locate and evaluate software packages may be obtained from several commercial directories. The following sources offer application software for sale.

--Computer manufacturers. While computer manufacturers have always offered system software (assemblers, compilers, operating systems, and so forth), some of them also offer assortments of application packages.

--Software houses. Companies that use computers are finding it increasingly difficult, and often economically impossible, to maintain in-house programming groups that are sufficiently large and competent to handle all their software development needs. As a result, they are increasingly looking toward outside suppliers. Many software houses still concentrate on contract programming tasks, in which the programs are custom-designed to meet each customer's specific needs. But an increasingly large number of software suppliers have recognized the potentially greater profits to be gained from "mass-producing" generalized packages, in which the development costs can be spread out over sales of multiple copies. Thus, if a sufficient number of copies can be sold, the supplier benefits from a higher total return on its development costs, while the customers benefit from a far lower price (typically 80 to 90 percent lower) than they would have to pay for similar custom-built programs.

--Software brokers. Several companies now act as brokers between software developers and buyers. Some of the wares are first-class packages developed specifically for sale to multiple users by independent software suppliers that lack the resources to market them nationally, but others are programs that were developed for use in a particular single installation and later "jury-rigged" for resale to others in the hope of recovering their development cost. Also, some of the brokers are fully staffed to install, support, and maintain the packages they sell, while others look to the original developer to perform these vital support functions. These additional considerations should be kept in mind when surveying the offerings of the software brokers.

--Turnkey system suppliers. These firms purchase appropriate computers and peripheral equipment, develop the necessary software, and supply the end user with a total system of hardware and software tailored to the particular requirements. This approach is very inviting to the nontechnical customer who wants the benefits of automating without the headaches of development. A reliable turnkey vendor that fully understands the user's computing needs, however, is necessary. It is also important that all of the software

installed by such a vendor be furnished in source-code form, in case the vendor goes out of business and the user is left to maintain it himself. Depending on the size of such a system, a delegation of procurement authority from GSA may be necessary.

--Computer stores. These recent arrivals in the software marketplace initially supported the computer hobbyists, but are moving into the small business area and offering "packaged" software. Use of this type of software is rarely warranted, and the customer must be very careful and skilled to select and use such software effectively.

GSA schedule contracts for software products

The GSA schedule contracts branch negotiates contracts annually with participating vendors. Each contract includes a price and certain terms and conditions, including the point in sales beyond which the Government attains a Government-wide license.

A product can be ordered from a vendor's schedule contract by writing a purchase order if the order does not exceed the contract's maximum order limitation. Also, effective January 15, 1981, agencies may procure software products separately from schedule contracts up to $50,000 from a sole-source vendor and up to $100,000 competitively, with no prior GSA approval. The separate contract type of software procurement requires notice of intent in the "Commerce Business Daily."

Government rights are protected and the basic services, prices, terms, and conditions are established in the schedule. For example, the agency must be satisfied with the delivered software, or the vendor must fix the problems within 30 days; otherwise, the agency may cancel the transaction. If an installation wants to negotiate for an extra service or try to get a better price, it can start with the schedule contract as a basis and negotiate a separate contract for its unique requirements. Vendors' addresses and telephone numbers are listed in the ADP Schedule.

Vendors do not need to spend the labor and experience the delay that would result if they had to negotiate a new contract with each Federal ADP installation. Since Federal customers can order software more quickly with schedule contracts, vendors will probably collect more quickly from sales under these contracts.

Two relevant documents are Federal Procurement Regulations Amendment 211, January 15, 1981, and Federal Procurement Regulation 1-4.11, "Procurement and Contracting for Governmentwide ADP Equipment, Software, Maintenance, Services, and Supplies."

Information is available at:

General Services Administration
Automated Data and Telecommunications Service
Procurement Division
ADP Schedule Contracts Branch
Washington, DC 20405
Telephone: (202) 566-1993

Evaluation factors that are not provided by GSA schedules but that must be considered

Once suitable packages are identified, criteria should be developed to compare and evaluate each package. Each evaluation factor may be weighted according to what is most important to the user in a particular situation. Some of the factors to be considered are as follows:

--Can the vendor demonstrate the package in an environment similar to the user's own?

--Can the vendor offer any training or installation services?

--Is the vendor suitably located to provide service if required?

--Are current users of the package available to attest to satisfactory performance? (Also check published user ratings.)

--Are the performance aspects (reliability, operating costs, and response times) of the software adequate? One source of performance data is current users with comparable equipment configurations and operational environments.

--Is the package flexible enough to accommodate changing requirements?

--Is the package delivered in object-language form only or is the source language provided for greater ease in understanding, utilizing, and modifying the package? (If object form only is delivered, then the customer should require that the source form be placed in escrow with a third party, for protection in case the seller goes out of business.

--Will the package interface with the necessary in-house existing systems?

--Is adequate documentation provided? Documentation should meet the needs of the systems analyst, the maintenance programmer, the computer operator, and the end user. Specific elements of documentation should include:

- Narrative descriptions
- System and program flowcharts
- Source program listings
- Input document layouts
- Internal records layouts
- Report layouts
- Operating instructions

The user documentation should be of special interest in the evaluation of the software package. Users need documentation for instructions and reference. The attention vendors give to user documentation is an indication of their professionalism and may reflect the quality of their overall product. A review of user documentation will give some indication of the cost to install the package. Ease of use of the documentation may be indicated by the following factors.

- Table of contents, index, and glossary.
- Sample problems and troubleshooting guide.
- Examples, illustrations, and other visual aids to legibility.

--What is the cost of acquiring and using the package? The cost should include any necessary modifications to the package.

--Is the vendor established and financially sound?

--Are there restrictions on the use of the package?

--Is there a warranty that provides for correction of defects and guarantees satisfaction with the package?

2. Using shared software

Factors used to determine the feasibility of software sharing are the same as those for buying software packages. Several catalogs showing available software are published by various State and Federal agencies. Also, individual inquiries may be made to agencies with similar missions.

The GSA Federal Software Exchange Center has a Government-wide mission for software sharing. It collects documentation abstracts and copies of source code from agencies that submit software for inclusion in the Center's catalog. The software is required to have been operational in the originating agency for 90 days. A requesting agency is given machine-readable source code and documentation. Neither installation assistance nor maintenance are now provided, but the software and documentation are available for far less than they would cost to develop in-house. Software is offered in 12 categories, including business and scientific applications and software tools. We found several very successful examples of agencies using this software at a fraction of what they would have paid had the software been developed new. Relevant documents include Standard Form 185, "Federal Information Standard Software Summary," and FSEC's "Federal Software Exchange Catalog."

The Center can be contacted at:

General Services Administration
Office of Information Resources Management
Office of Software Development
Federal Software Exchange Center
2 Skyline Place, Suite 1100
5203 Leesburg Pike
Falls Church, VA 22041
Telephone: (703) 756-6153

Evaluating software applications already owned by another Government agency and available under a sharing arrangement is much the same as evaluating a vendor package. The following factors, however, do make a slightly different approach necessary.

--The cost of a copy of the unmodified application is almost always less than the purchase price of a vendor package.

--The software may be difficult to modify, either technically for another computer system, or functionally for another user organization.

--If modifications, extra documentation, or assistance are needed, a reimbursable agreement may have to be worked out with the originating organization. Such modification can add significantly to the "copying costs" of sharing the software.

--The amount of implementation assistance and future support of the application that is available from the originating organization may be a factor.

Most of the other evaluation factors, such as the quality and suitability of the software, and hardware compatibility, are the same as for vendor packages.

3. Enhancing an organization's existing software

Here we mean incremental improvements of software that is already present in the organization. This method has two advantages:

--Existing applications can provide templates against which the improvements can be tested, for example, machine readable files of test data and the results therefrom.

--Employees will probably resist it less than they would outside software.

Some considerations argue against this method:

--Some organizations' present software is so bad that enhancing it would be as much trouble as new development. It was

developed wih old languages or technologies, maintained without discipline for years, documented poorly or not at all, and has long since been abandoned by its authors.

--Enhancement efforts tend to be postponed.

--Costs of in-house enhancement are most difficult to identify.

--Technology and skills must be present to make software enhancement feasible.

Software tools can help software evaluation, modification, and testing

Evaluating and, if need be, modifying and testing existing application software (from any source) can be greatly helped by using special computer programs called software tools. Software tools are computer programs that manipulate other computer programs and aid their producton, inspection, modification, and/or testing.

A software tool may itself be written in a higher level language, which may be the same language as that used for the application programs to be manipulated. However, a number of commercially available software tools are written in lower level, machine-dependent assembly language and thus will operate only on a machine architecture that will work with the specific assembly language used. Thus, commercially available software tools are often operable only on IBM and compatible architectures--vendors orient their products to IBM because IBM and compatible machines dominate the commercial market. However, the Federal Government's computers are mostly non-IBM and thus cannot operate IBM-dependent software tools. Since the Government's non-IBM sites have fewer software tools, fewer opportunities for software productivity enhancement are available to them than to the IBM sites.

Our report on software technology[3] recommended that the Administrator of General Services:

"* * * Establish, by development or adoption, a set of standard tools and methods to solve operational problems, promote efficiency and economy, and inspect software * * * "The tools adopted should themselves be written in higher-level languages, where possible, to maximize their portability to different brands of computers."

In April 1982, GSA's Federal Software Testing Center (until recently called the Compiler Testing Center) published a catalog of

---

[3]Wider Use of Better Computer Software Technology Could Increase Management Control and Reduce Cost," FGMSD-80-38, Apr. 29, 1980, pp. 38 and 39.

software tools.[4] The catalog offers four tools for COBOL, including test coverage monitor, a source code formatter, a file compare utility, and a cross-reference documentation tool, and one tool for FORTRAN. The COBOL tools are themselves written in COBOL, making them transportable to any brand of computer that supports full ANSI-74 COBOL or full ANSI-68 COBOL (for older installations). The tools are offered with support of the same type provided with vendor products; that is, the Testing Center will provide technical help with installing and using the tools on a reimbursable basis.

III. SELECTING ALTERNATIVES

All viable alternatives should be compared and the most suitable one chosen based on economic considerations when all other factors are equal. Factors should be weighted, however, to reflect critical requirements in a given situation. Long range plans of the agency, such as the scheduled addition or deletion of functions, a known future change in computer hardware, and other factors that the selection is likely to affect, should be considered.

The selection can be dictated by pressures that really leave only one choice, including:

--Small projects will be done in whatever way is convenient because (1) they are too small to justify a search for better ways and (2) there may not be time to find a better way.

--There may be employees available to use the methods the organization now has and no money available for either training in new methods or for procuring software.

--A legislated deadline may drive developers to use existing software because there is simply no time for developing new software.

---

[4]"Software Tools Catalog," Report FCTC-82/013 (Falls Church, Va., Apr. 1982).

RELEVANT DOCUMENTS

A. LAWS, REGULATIONS, AND OMB CIRCULARS

Laws

Public Law 89-306, "An Act to provide for the economic and efficient purchase, lease, maintenance, operation, and utilization of automatic data processing equipment by Federal departments and agencies," Oct. 1965 (the Brooks Act).

Public Law 96-511, "An Act to reduce paperwork and enhance the economy and efficiency of the Government and the private sector by improving Federal information policymaking, and for other purposes," Dec. 11, 1980 (the Paperwork Reduction Act).

Regulations

Federal Procurement Regulations Amendment 211, Dec. 29, 1980, General Services Administration, Washington, D.C. 20405.

Federal Procurement Regulations Subpart 1-4.11, "Procurement and Contracting for Governmentwide Automated Data Processing Equipment, Software, Maintenance Services, and Supplies," General Services Administration, Washington, D.C. 20405.

Federal Property Regulations Subpart 1-4.12, Amendment "ADP Services Contracts," May 12, 1982.

Bulletins

GSA Bulletin FPMR F-131 ADP and Telecommunications, "Contracting for Software Development," May 19, 1981, General Services Administration, Washington, D.C. 20405.

GSA Bulletin FPR F-51 Federal Procurement, "Contracting for Software Development," May 19, 1981, General Services Administration, Washington, D.C. 20405

OMB Circulars

Circular No. A-121, "Cost Accounting, Cost Recovery, and Inter-Agency Sharing of Data Processing Facilities," Sept. 16, 1980, Office of Management and Budget, Washington, D.C. 20503.

B. REFERENCES

General Accounting Office

"Federal Agencies' Maintenance of Computer Programs: Expensive and Undermanaged," AFMD-81-25, Feb. 26, 1981.

"Government-Wide Guidelines and Management Assistance Center Needed To Improve ADP Systems Development," AFMD-81-20, Feb. 20, 1981.

"Non-Federal Computer Acquisition Practices Provide Useful Information for Streamlining Federal Methods," AFMD-81-104, Oct. 2, 1981.

"Wider Use of Better Computer Software Technology Can Improve Management Control and Reduce Costs," FGMSD-80-38, Apr. 29, 1980.

"Contracting for Computer Software Development--Serious Problems Require Management Attention To Avoid Wasting Additional Millions," FGMSD-80-4, Nov. 9, 1979.

"Shifting the Government's Automatic Data Processing Requirements to the Private Sector: Further Study and Better Guidance Needed," FGMSD-78-22, Apr. 11, 1978.

"Millions in Savings Possible in Converting Programs from One Computer to Another," FGMSD-77-34, Sept. 15, 1977.

"The Federal Information Processing Standards Program: Many Potential Benefits, Little Progress, and Many Problems." FGMSD-78-23, Apr. 19, 1978.

"The Federal Software Exchange Program--A Small Step in Improving Computer Program Sharing," FGMSD-78-11, Jan. 13, 1978.

"Guidelines for Accounting for Automatic Data Processing Costs," Federal Government Accounting Pamphlet Number 4, 1978.

"Acquisition and Use of Software Products for Automatic Data Processing Systems in the Federal Government," B-115369, June 30, 1971.

General Services Administration

Office of Software Development, Automated Data and Telecommunications Service, "Software Improvement--A Needed Process in the Federal Government," Report OSD-81-02, June 3, 1981.

GSA/ADTS/C-81/1, "Federal Software Exchange Catalog," PB 81-904001, Jan. 1981.

Federal Software Testing Center, "Software Tools Catalog," Report FCTC-82/013, Falls Church, Va., Apr. 1982.

Federal Compiler Testing Center, "A Software Tools Project: A Means of Capturing Technology and Improving Engineering," Report OSD-82-101, Feb. 1982.

National Bureau of Standards

Adrion, W. Richards, Martha A. Branstad, and John C. Cherniavsky, "Validation, Verification, and Testing of Computer Software," NBS Special Publication 500-76, Feb. 1981.

Houghton, Raymond C., Jr., and Karen A. Oakley, eds., "NBS Software Tools Database," NBSIR-80-2159, Oct. 1980.

Collica, Joseph, Mark Skall, and Gloria Bolotsky, "Conversion of Federal ADP Systems: A Tutorial," NBS Special Publication 500-62, Aug. 1980.

Adrion, W. Richards, Martha A. Branstad, and John C. Cherniavsky, "Validation, Verification, and Testing for the Individual Programmer," NBS Special Publication 500-56, Feb. 1980.

"Guidelines for Documentation of Computer Programs and Automated Data Systems for the Initiation Phase," FIPS PUB 64, Aug. 1, 1979.

"Guide to Computer Program Directories," NBS Special Publication 500-22, Dec. 1977.

Dennis W. Fife, "Computer Software Management: A Primer for Project Management and Quality Control," NBS Special Publication 500-11, July 1977.

"Guideline on Computer Performance Management: An Introduction," FIPS PUB 49, May 1, 1977.

Deutsch, Donald R., "Appraisal of Federal Government COBOL Standards and Software Management: Survey Results," NBSIR, 76-1100, Aug. 1976.

"Guidelines for Documentation of Computer Programs and Automated Data Systems," FIPS PUB 38, Feb. 15, 1976.

"Aids for COBOL Program Conversion," FIPS PUB 43, Dec. 1975.

"COBOL," FIPS PUB 21-1, Dec. 1, 1975.

Office of Management and Budget

President's Reorganization Project Report, "Information Technology and Government Reorganization: Summary of the Federal Data Processing Reorganization Project," Apr. 23, 1979.

Other Sources

General Accounting Office, Joint Financial Management Improvement Program, "Do It Yourself--Compare and Improve Your Payroll System," Apr. 29, 1981.

"ICP Interviews Werner L. Frank," Interface, Winter 1980.

"The Make-or-Buy Decision," Datapro, Nov. 1979.

"Software Vendor Support Survey," Datapro, April 1980.

EDP Weekly, Nov. 1980.

Martin, A., "The Software Package Generation," Computerworld, Oct. 1980.

"How to Buy Software Packages," Datapro, Mar. 1979.

Frank, Werner L., "The Ten Great Software Myths," Computerworld.

Hoard, Bruce, "Rising Software Costs Found Offset by Other Savings," Computerworld, Dec. 1980.

"Guide to Software Products," Datapro, Nov. 1979.

"A Review of Software Cost Estimation Methods," Datapro, May 1978.

"An Overview of the Make Vs. Buy Decision," Datapro, Nov. 1979.

APPENDIX II

AGENCY COMMENTS

UNITED STATES DEPARTMENT OF COMMERCE
The Inspector General
Washington, D.C. 20230

October 18, 1982

Mr. Henry Eschwege
Director, Community and Economic
Development Division
U.S. General Accounting Office
Washington, D. C. 20548

Dear Mr. Eschwege:

This is in reply to your letter of September 7, 1982, requesting comments on the draft report entitled "Federal Agencies Could Save Time and Money with Better Computer Software Alternatives."

We have reviewed the enclosed comments of the Director, National Bureau of Standards and believe they are responsive to the matters discussed in the report.

Sincerely,

Sherman M. Funk
Inspector General

Enclosure

APPENDIX II

Comments of

National Bureau of Standards
Department of Commerce

on

GAO Report to the Congress

on

"Federal Agencies Could Save Time and Money
With Better Computer Software Alternatives"

APPENDIX II

COMMENTS ON RECOMMENDATIONS TO THE SECRETARY OF COMMERCE

The proposed report is timely and addresses a very important set of issues associated with software development and use. It should draw attention to the major benefits possible through reuse of existing software, especially software that is commercially developed and supported.

This report is consistent with the current program of the Institute for Computer Sciences and Technology (ICST) at NBS. ICST is developing guidelines for Federal agencies on the specification, evaluation, selection, and testing of applications software packages.

General
Services
Administration Washington, DC 20405

OCT 14 1982

Honorable Charles A. Bowsher
Comptroller General of the United States
U.S. General Accounting Office
Washington, DC 20548

Dear Mr. Bowsher:

The General Accounting Office (GAO) draft report entitled, "Federal Agencies Could Save Time and Money with Better Computer Software Alternatives," contains recommendations to the Administrator of General Services. We thank you for the opportunity to respond to those recommendations.

We agree with GAO that alternatives to custom development for meeting agencies' software needs are essential to reducing the billions of dollars spent annually on software. The Office of Software Development (OSD), formed by GSA in May 1980, has the mission of reducing these costs and improving the service levels of this expenditure.

The report cites three alternatives to the traditional software development: use of already-existing software (either buying or sharing), use of more convenient ways of developing new software, and use of prefabricated software components. We believe that any one or a combination of all three of these alternatives can be used to meet software requirements through software improvement projects. This allows incremental evolution of a system to meet the changing needs of an agency and incorporates the latest technology while past investment is preserved as much as possible. Software improvement projects allow for the transfer and integration of existing code, other operational code, and new code into the new system. A copy of "Software Improvement: A Needed Process in the Federal Government," prepared by OSD, is enclosed for your information.

GSA supports the two recommendations to the Office of Management and Budget (OMB). The combining of all software sharing activities would provide a single source for federally-owned software available for exchange and reduce the time and effort involved in locating a required piece of software. GSA has the framework in place to act as an agent for other exchanges with minimal additional resources. GSA concurs with the recommendations to Heads of Agencies.

In response to the recommended actions for the Administrator of General Services, we offer the following.

1. Recommendation: The ADP Schedule Contracts Branch of GSA's Office of Information Resources Management require vendors to complete a standard Software Summary (SF-185) on each software product for which they negotiate a contract and forward the summaries to the GSA Federal Software Exchange Center for inclusion in the catalog section that deals with vendor software, to better inform agencies about what is available.

Response: When a vendor is awarded an ADP Schedule contract, a letter is sent to the vendor informing him of the opportunity to list his product(s) in the Federal Software Exchange Program (FSEP) Catalog and requesting appropriate input. The response to our request was surprisingly small during FY 1982. To encourage a greater response, a new letter, stressing the benefits of listing these ADP Schedule products in the FSEP Catalog, was sent to all vendors negotiating contracts for FY 1983. Making listings mandatory, however, would impose another unwanted reporting requirement on the private sector.

GSA will implement the recommendation to make the ADP Schedule price lists available to senior Federal officials. For the FY 1984 contract period, we will provide a mailing list of the senior ADP official and the senior procurement official in each agency to our software contractors for mailing their ADP Schedule price lists. However, the contractors will not be required to provide their price lists to these officials.

2. Recommendation: GSA's Office of Software Development demonstrate the concept of Federal use of vendor-developed application software by selecting from one to three vendor developed application software packages and by modifying them for general use by Federal agencies as pilot projects. Payroll, inventory, and case control might be suitable candidates.

Response: GSA assumes that "vendor-developed application software" referred to vendor-developed proprietary application software; and, we have based our response to this recommendation on the assumption.

Although we fully support the use of vendor proprietary software, we do not feel this recommendation is practical. Obtaining rights to modify and distribute a proprietary package would be very expensive because it would preclude any further sales to the Government. Other complicating factors would be GSA maintenance of the modified proprietary package and obtaining new versions of the package should they be developed by the vendor.

APPENDIX II

In support of wider use of the proprietary packages throughout the Federal Government, GSA's Office of Software Development has awarded two contracts to determine the reasons for the apparent reluctance of the vendors to market to the Federal Government and to determine why agencies resist purchasing off-the-shelf, tested software. The results of these surveys will be used to implement new techniques to stimulate vendor interest in the Federal market and to encourage agencies to use such software. We believe that once the vendor and agency problems and attitudes are better understood, we can develop more cost-effective techniques to increase the use of packaged software rather than to assume the role of developers ourselves.

3. Recommendation: GSA's Federal Software Exchange Center demonstrate the concept of deliberate reuse of federally owned application software by acquiring, enhancing, and advertising for general use at least one commonly used large scale application, such as a Federal personnel system, as a pilot project.

Response: We agree with the intent of the recommendation; but, believe that there are better ways to implement it. This function can and has been successfully accomplished by private industry. One Washington based company devotes almost its entire staff to modifying, enhancing, and providing assistance in implementing a federally-owned case tracking system. Although this system was designed for State and local government legal case tracking, it has been modified to track inmates in jail, parcels of land, tort cases in New York State, and is in use in all 94 US Attorney's offices and several other Federal agencies. This system could be further modified to track welfare recipients or any function requiring tracking. The vendor charges only for the value added modifications, implementation, and maintenance.

It should be recognized that unless software was specifically designed as a framework system, it is very difficult to modify for general use. A better approach is to develop framework systems when a specific system is needed by an agency. OMB has an opportunity during tne budget review and the review of agency long range plans to identify agencies planning to obtain software for common functions. (The Department of Treasury, for example, recently requested funds for the development of a new payroll/ personnel system.) In each case, additional funds could be allocated by OMB to develop a single system with transportability as

an initial design goal. Once developed, the system could be centrally maintained and offered to other agencies as a framework system much more easily modified to meet other prospective user needs.

Sincerely,

Ray Kline
Deputy Administrator

Enclosure

EXECUTIVE OFFICE OF THE PRESIDENT
OFFICE OF MANAGEMENT AND BUDGET
WASHINGTON, D.C. 20503

OCT 18 1982

Mr. W. D. Campbell
Acting Director, Accounting and
Financial Management Division
General Accounting Office
Washington, D.C. 20548

Dear Mr. Campbell:

This in in response to your letter of September 3, 1982 requesting OMB comments on a draft report entitled "Federal Agencies Could Save Time and Money with Better Computer Software Alternatives."

The report has two basic thrusts: (1) to encourage sharing; and (2) to educate Federal software consumers as to the availability of commercial software products.

While we agree with the general approach recommended in the report, we have some reservations about software sharing as an effective and efficient alternative. There are great risks in the wholesale transfer of software not designed with portability in mind. If the developing agency did not design, develop, and document the system with transferability as an objective, adaption and maintenance costs may exceed even those of custom built software. There have been some successes in software transfer; e.g., the adoption by several civilian agencies of the Air Force civilian personnel system, but we would urge caution and selectivity.

There are a number of other activities under way that will impact on this area. As a part of our Reform 88 initiative, we are looking at the possibility of designating or developing central service centers for common application requirements; e.g., payroll or check processing. That approach would be more cost effective than maintaining dozens of individual agency systems to perform like functions.

We are also looking at ways to change the incentives that presently encourage Federal managers to prefer custom-built to commercially available software solutions. One approach being considered is to shift more decision responsibility for use of information technology to the program official who uses it.

Recent initiatives to rid the Federal inventory of obsolete hardware should also have a salutory effect on the software problem. Some of the oldest and most difficult to maintain custom-built Federal software systems will come under scrutiny as agencies move to more modern hardware. Since the commercial

APPENDIX II

market provides a wider range of software alternatives for more modern equipment, we would expect an increase in use of commercial packages would result.

With respect to the specific recommendations in the draft report:

Recommendations

"The Director of the Office of Management and Budget [should] direct that:

"- Agencies which develop applications make their software and documentation available to FSEC.

"- Agencies which operate software sharing activities of their own make software, documentation, and directories available to FSEC."

OMB Comment

We have no objections to encouraging agencies to follow these recommendations with the understanding that, as suggested above, sharing is but one, and probably not the preferred, alternative to custom software development.

Recommendation

"- OMB [should] analyze the possibilities of combining other agencies' software sharing efforts with those of FSEC to reduce duplication as part of OMB's general charter to designate collection points and reduce duplication under Public Law 96-511, the Paperwork Reduction Act. If the analysis shows that such combination would save money, OMB should direct action; if not, the reasons why should be made public."

OMB Comment

We concur in this recommendation to the extent that agencies are operating duplicative clearinghouse functions, their activities should be coordinated.

We appreciate the opportunity to comment on this report and the efforts of GAO staff to identify the causes of and prescribe remedies for rising Federal software maintenance costs. We look forward to continuing close cooperation in this endeavor.

Sincerely,

Joseph R. Wright, Jr.
Deputy Director

(913663)

25201

AN EQUAL OPPORTUNITY EMPLOYER

UNITED STATES
GENERAL ACCOUNTING OFFICE
WASHINGTON, D.C. 20548

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE,$300

POSTAGE AND FEES PAID
U. S. GENERAL ACCOUNTING OFFICE

THIRD CLASS