

DOCUMENT RESUME

03389 - [A2593734]

Millions in Savings Possible in Converting Programs from One Computer to Another. FGMSD-77-34; B-115369. September 15, 1977. 14 pp. + 3 appendices (45 pp.).

Report to the Congress; by Elmer E. Staats, Comptroller General.

Issue Area: Automatic Data Processing; Changeover to Other ADP Systems (107).

Contact: Financial and General Management Studies Div.

Budget Function: Miscellaneous. Automatic Data Processing (1001).

Organization Concerned: Department of Commerce; Department of Defense; General Services Administration; Office of Management and Budget.

Congressional Relevance: Congress.

Authority: Brooks Act (P.L. 89-306).

Frequently, computer programs must be converted to make them run on a computer different from the one for which they were originally devised. The annual Federal cost of such conversions is estimated at more than \$450 million; over \$100 million of this could be saved. Findings/Conclusions: The following factors tend to increase conversion costs: lack of readily available software conversion expertise in the Government; poor quality of software to be converted; inadequate flowcharts and other explanatory data; selection of nonstandardized equipment; and lack of programmer productivity aids. The Navy and the General Services Administration (GSA) are planning a Federal center for software conversion. This center could provide the know-how that is needed in software conversion and make it readily available to all agencies. Recommendations: The Office of Management and Budget should assist in establishing a Federal center for software conversion. When such a center is established, GSA should be able to require independent conversion estimates by the center for expensive agency procurements which have significant conversion costs. The center could also assist agencies in converting programs collected by GSA's recently established Federal Software Exchange Center. Heads of Federal agencies should emphasize quality and standards in new software development, including both the programs themselves and their documentation. The National Bureau of Standards should select and publish a set of programmer productivity aids for Government-wide use to improve the productivity of computer programmers on both original development of software and software conversion. (Author/SC)

3734

# REPORT TO THE CONGRESS

BY THE COMPTROLLER GENERAL  
OF THE UNITED STATES

---

03389



## Millions In Savings Possible In Converting Programs From One Computer To Another

Office of Management and Budget  
National Bureau of Standards

Frequently, computer programs must be converted to make them run on a computer different from the one for which they were originally devised. The annual Federal cost of such conversions is estimated at more than \$450 million. GAO estimates that over \$100 million of this could be saved. Further savings will depend on future developments, some technological in nature.



COMPTROLLER GENERAL OF THE UNITED STATES  
WASHINGTON, D.C. 20548


B-115369

To the President of the Senate and the  
Speaker of the House of Representatives

This report describes (1) the problems Federal agencies have in converting computer programs from one computer to another, (2) the annual cost and savings possible in this activity, and (3) how the savings can be realized.

We made our review pursuant to the Budget and Accounting Act, 1921 (31 U.S.C. 53), and the Accounting and Auditing Act of 1950 (31 U.S.C. 67).

We are sending copies of the report to the Director, Office of Management and Budget; the Secretaries of Commerce and Defense; and the Administrator of General Services, who are addressed specifically by our recommendations, and to other interested parties.

  
Comptroller General  
of the United States

COMPTROLLER GENERAL'S  
REPORT TO THE CONGRESS

MILLIONS IN SAVINGS POSSIBLE  
IN CONVERTING PROGRAMS FROM ONE  
COMPUTER TO ANOTHER  
Office of Management and Budget  
National Bureau of Standards

D I G E S T

In the early days of computers, the price of the equipment was the largest part of the costs. The programs which made the equipment operate cost relatively little. Today the programs--software as they are called in the data processing community--cost considerably more than the equipment in most systems.

One part of software cost that is large and not directly productive is called conversion cost. This is incurred to make programs devised for one computer work on another computer of a different make or model. Conversion costs should be reduced.

If you buy a new stereo phonograph, you expect your present library of stereo records to play on it. This is because turntable speeds and record grooves are the same on all makes of phonographs. You cannot expect such a carryover where computers and computer programs are concerned. There is little standardization in computers; even different models of the same make may not run each other's programs without some changes.

Considerable work is often necessary to make old programs "play" on a new computer. Admittedly, this analogy is simplistic--computers and phonographs differ greatly in complexity--but it gives an idea of the situation. The effort needed to make computer programs work on a new computer varies greatly from case to case but, in the aggregate, is quite large.

No good figures are available on Federal conversion costs as such, but GAO estimates them at about \$450 million a year. GAO estimates that about 24 percent--over \$100 million--could be avoided in today's environment. (See pp. 3 and 4.)

FGMSD-77-34

The following factors tend to increase conversion costs. (See pp. 5 to 7.)

- Lack of readily available software conversion expertise in the Government.
- Poor quality of the software to be converted, including unnecessary use of features peculiar to an old computer which do not work on the new one.
- Inadequate flowcharts and other data (called documentation) which explain the working of the software to be converted.
- Selection of new computer equipment that requires more work on conversion because equipment has not been standardized among manufacturers.
- Lack of "programmer productivity aids," or a set of tools and techniques to be used by programmers, which could reduce the amount of human effort needed in conversion.

Not all of these factors are readily controllable, but GAO estimates that costs in the current environment can reasonably be reduced about 24 percent with good conversion planning and practices. Since such a reduction offers savings estimated at over \$100 million a year and because better conversions would result in prompt-er, more effective Government services, the work needed to bring about better conversions, although considerable, will be worthwhile.

The Navy and the General Services Administration are planning a Federal center for software conversion. (See app. I). The center could provide the know-how that is needed in software conversion and make it readily available to all agencies.

To reduce conversion costs, GAO recommends that (see pp. 11 and 12):

- The Office of Management and Budget assist in establishing a Federal center for software conversion. When such a center is established, the General Services Administration should be able to require independent conversion estimates by the center for expensive agency procurements which have significant conversion costs. The center could also assist agencies in converting programs collected by General Services Administration's recently established Federal Software Exchange Center.
  
- Heads of Federal agencies emphasize quality and standards in new software development, including both the programs themselves and their documentation.
  
- The National Bureau of Standards select and publish a set of programmer productivity aids for Government-wide use to improve the productivity of computer programmers on both original development of software and software conversion.

Several agencies provided written comments on the matters discussed in this report. They generally agreed with GAO's conclusions and recommendations. (See pp. 12 to 14.)

## C o n t e n t s

		<u>Page</u>
DIGEST		i
CHAPTER		
1	INTRODUCTION	1
	Roles of various agencies	1
	Scope of review	2
2	SOFTWARE CONVERSION IS A FREQUENT AND COSTLY UNDERTAKING	3
	Why a better approach to software conversion is needed	3
	Problems in software conversion	4
	Ways in which software conversion costs can be reduced	7
3	CONCLUSIONS, RECOMMENDATIONS, AND AGENCY COMMENTS	10
	Conclusions	10
	Recommendations	11
	Agency comments	12
APPENDIX		
I	Where computer software is now, and how it got there	15
II	Details of the separate sources of in- formation	27
III	Agency comments	43

### ABBREVIATIONS

ADP	automatic data processing
COBOL	Common Business Oriented Language
DOD	Department of Defense
GAO	General Accounting Office
GSA	General Services Administration
HEW	Department of Health, Education, and Welfare
NBS	National Bureau of Standards
OMB	Office of Management and Budget

## CHAPTER 1

### INTRODUCTION

This report discusses the conversion of applications software, including its costs, causes of problems, and possible improvements. Applications software automates the tasks of users--such as payroll, accounting, and statistical calculations. Software conversion is the act of making computer programs run on some computer other than the one for which they were originally devised. The reason for converting existing application programs, instead of writing new ones for the new computer, is that it is usually quicker and cheaper. Application programs are converted when (1) a replacement computer has been acquired and (2) where it is desired to share a program that was written at another place. Recent congressional hearings 1/ were concerned with the treatment of conversion costs in automatic data processing (ADP) procurement, but this report deals with conversion matters in general, not procurement policy alternatives.

### ROLES OF VARIOUS 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 procuring and maintaining Federal ADP resources. GSA receives technical advice from the Secretary of Commerce primarily through the National Bureau of Standards (NBS) and both these agencies receive fiscal and policy guidance from the Office of Management and Budget (OMB).

In our role of aiding the Congress, we are concerned with the management of Federal ADP and with computer software conversion as a frequent and expensive activity. 2/ Our past reports to the Congress have recommended improvements in ADP management on a Government-wide basis and at specific agencies.

---

1/House Committee on Government Operations "Administration of Public Law 89-306, Procurement of ADP Resources by the Federal Government," October 1, 1976.

2/For a discussion of the present state of computer applications programs, see app. 1.



## SCOPE OF REVIEW

Our review included questionnaires to Federal programmers and supervisors <sup>1/</sup>, interviews with experts, and a study of the literature.

The questionnaires were administered to 2,005 programmers and 465 supervisors of ADP functions at 43 Federal computer installations, including business and scientific ones. Twenty-six were Department of Defense (DOD) installations and 17 were not. The purpose of the questionnaires was to obtain information on Federal conversion problems and remedies and cost estimating data from working level employees directly involved in the process.

We discussed conversion problems with recognized software experts from both the Government and the private sector. We also made an extensive study of literature on software in general and software conversion in particular.

Our review yielded an extensive bibliography, a software glossary, and a provisional planning checklist for software conversion projects. All are available on request. For copies, write to:

U.S. General Accounting Office  
Attn: FGMSD-ADP  
441 G Street NW.  
Washington, D.C. 20548

-----  
1/See app. II for some detail of our sources.

## Chapter 2

### SOFTWARE CONVERSION IS A FREQUENT AND COSTLY UNDERTAKING

Software conversion is a frequent and expensive part of Federal ADP operations. Replacement computers are bought and installations must convert to them. Also, programs that are shared must be modified before they will work on the receiving installations' systems. Each situation requires a conversion effort which can range from trivial to enormous.

### WHY A BETTER APPROACH TO SOFTWARE CONVERSION IS NEEDED

To assess the importance of software conversion, we asked about its frequency both in our questionnaires and our discussions with experts. We found that conversion to replacement systems is very common in Federal agencies, although at any one installation 7 years is the average time between computer system replacements. Federal organizations contacted during this review cited conversion problems as a major concern. On our questionnaires, over three-fourths of the supervisors and over two-thirds of the programmers indicated that they had worked on at least one conversion to a replacement system, while one-half of the supervisors and one-third of the programmers indicated that they had been involved in at least one conversion of a program brought in from another installation.

The Federal cost of software conversion for all ADP categories is estimated to be over \$450 million annually. Its potential for reduction in today's environment is estimated to be over \$100 million annually. This saving is available solely from doing the same jobs better in the current environment. Other alternatives, such as buying or sharing software, also have a great potential for more software cost avoidance, but such savings cannot now be quantified.

We know of no good figures available on what the conversion of computer programs costs the Government each year or how much of it could be saved by better conversion techniques. Consequently, we could not make a precise determination of what either the costs or the potential savings might be. Recognizing, however,

that estimates are often useful in judging whether proposals are worthwhile, we made the best estimates we could of costs and potential annual savings. Admittedly, they are based on many approximations, but we believe they are reasonable, and they are supported by several consultants' estimates of conversion costs which were about the same as ours.

In making our estimate, we determined that the Federal Government spends about \$6 billion a year on software for all types of ADP and that about half of this, or \$3 billion, is spent for maintaining and converting computer programs after the software is acquired. We estimated that conversion costs to replacement systems are one-seventh of the \$3 billion, or over \$425 million, using 7 years as the typical life of a hardware system and 1 year as the typical time it takes to completely convert an installation. Another \$25 million was estimated for converting programs acquired elsewhere (foreign programs) for use on installation computers. The \$25 million estimate is based upon programmers' estimates of the percentage of their time they spend on converting foreign programs, the number of programmer and systems analyst staff-years reported to GSA, and the estimated average cost of a software staff-year. Thus, a total of over \$450 million is arrived at for converting computer programs.

To get an idea of how much of this amount could be saved by better techniques, we obtained estimates from programmers and their supervisors through our questionnaire and, separately, from experts we contacted in the fields of computer program development and software conversion. The programmers and their supervisors estimated that over 40 percent could be saved, while the experts' estimates averaged about 24 percent. We believe the experts, with their broader experience in the complexities of overall design and conversion work, have a better total perspective of the problem, and their more conservative estimate is probably more realistic. Using the experts' percentage, we estimated the potential savings in conversion costs in the current environment as over \$100 million. Our suggestions for achieving these savings are discussed later in this chapter.

## PROBLEMS IN SOFTWARE CONVERSION

### General costs

Software conversion costs are incurred in several ways, including delays in user tasks, labor, retraining, inter-

ference, morale, and documentation. Each is discussed briefly below.

- Delay in automating user tasks. This cost of the conversion process is difficult to estimate but may be the most important in some cases. If a conversion project is late, the user does not get the task(s) automated at the expected time. Hence, the adverse impact on the user's work or his clients may be great.
- Programmer and analyst labor. This is the major direct cost of conversion. We estimated that a typical staff-year of this type, with overhead and computer support, cost \$33,000 in fiscal year 1976.
- Retraining. This cost, which will be greater when the conversion is to a replacement system of a different vendor, includes retraining of system programmers, computer operations staff, applications programmers, users, and management. The retraining needed in the case of converting a program from another installation depends entirely upon specifics of the situation, including the quality and size of the program being converted, the degree of communication between originator and recipient, and the knowledge and motivation of the responsible programmers at the receiving installation.
- Interference with other uses of the computer. A lengthy conversion effort may seriously interfere with other work that is scheduled to be done on the computer at the same time. On the other hand, the conversion could suffer if other work has priority.
- Employee morale. Employee morale may suffer during a conversion, especially if (a) there are indications of poor planning or (b) prolonged overtime is involved.
- Changing documentation. This cost is very situation dependent. If original documentation is good and kept in a machine-readable form, it will be easier to change, but if documentation is poor, significant time and effort could be required to change it.

Software conversion problems identified in our review

- Lack of readily available software conversion expertise. Both the questionnaire respondents

and the experts identified the lack of Federal software conversion experts and knowledge as the most important causes of problems in converting programs from other installations and significant causes of problems with conversions to replacement systems. Experts cited the fact that the typical installation which performs a conversion with its own staff is using people who very rarely do conversions since computers are generally replaced only every 5 to 10 years. The experts said that conversion would be better done by specialists. A Federal conversion center was often suggested as a solution.

- Poor quality of the software to be converted to replacement systems and of its documentation. Since programs are usually converted by persons other than their authors, either aspect of quality can vastly complicate the conversion. Poor quality of existing software has resulted from management failures to require excellence in software development and documentation, and to consider eventual conversion in the development of new software. Poor quality of the software includes code that is needlessly complex and difficult to understand and that uses unnecessary computer resources. Failure to consider eventual conversion results in programmers being allowed to unnecessarily use features of the current computer's programming languages that will not work on another make.
  
- Lack of standardization among manufacturers. Manufacturers typically offer ADP systems and programming languages with features which are unique to their equipment. In the past, programmers have been frequently allowed to, and have sometimes needed to, use such features. Thus, an installation's software inventory typically includes programs using such unique features, which do not conform to standards. If the replacement system is made by a different vendor, the unique features of the original vendor usually will not work on the replacement system, and partial reprogramming is necessary to make them work. Better adherence to standards in software development to reduce the use of vendor-unique features would reduce this problem.

--Lack of staff conversion experience, knowledge, or interest. This problem results because a given installation very seldom experiences conversion to a new system. (A 7-year system life is often cited.) Thus, due to normal turnover, many of the employees who worked on a prior conversion may not be there for the next one. Also, people who are very skilled on one manufacturer's systems may know nothing of another's. Another staffing problem is that many programmers and analysts are not interested in conversion. They perceive developing new software as more challenging and creative than conversion.

--Lack of programmer productivity aids. Programmer productivity aids can automate much of the drudgery of both writing original programs and conversion. Some can even provide partial translation from one manufacturer's programming language to that of another. If these aids are not present and used, programmers must do much more by hand.

The problems identified for conversion of programs from other installations were basically similar to those identified for conversion to replacement systems. They included poor quality of the material to be converted and the conversion staff's lack of knowledge. However, there was the added problem of programs from other installations not being adequately studied before the decision was made to bring them in to see if they matched the needs of the recipient. In these cases, there was not only a conversion effort but also a modification effort much larger than if more suitable programs had been chosen. This problem indicates that careful study is needed before the decision is made to share programs from other installations.

#### WAYS IN WHICH SOFTWARE CONVERSION COSTS CAN BE REDUCED

There are several ways to improve software conversion which apply generally to both types of conversion.

It would be helpful, for agencies anticipating a conversion project, to be able to turn to recognized impartial experts who could aid them in avoiding the common pitfalls. We believe that significant savings would result if a center of expertise were available to such agencies to assist with conversions. The expert assistance

would include impartial evaluation of conversion aspects of procurements, selection of contractors to perform conversions, project management of the teams actually doing the conversions (either agency or contractor), and transfer of lessons learned from one conversion to another and from one agency to another. Such a center could also actually do conversions--requiring a larger staff than if conversions were not done by the center--and assist agencies that want to share software from the Federal Software Exchange Center.

The Navy and GSA are now developing a cooperative agreement to establish such a center. We believe that such action would be a step in the right direction.

More emphasis on quality in the original development of software and documentation would ease eventual conversion problems. Means of attaining better quality include better enforcement of documentation and other standards, adoption of the new software technologies 1/, better training for programmers and systems analysts, and adoption of productivity aids.

The possible greater cost of higher quality software development would be more than offset by (1) easier and cheaper future conversion and modification of the software, (2) fewer errors in the supported user tasks, and (3) more feasibility of sharing the software by multiple users. Also, the productivity improvements available from the new programming technologies and cooperative efforts to develop software for groups of users indicate that higher quality software need not necessarily cost more, and may in fact be cheaper, than the old method.

More widespread use of automated programmer productivity aids would also ease software conversion and original software development problems. These aids, which are already sold in the marketplace, include translation programs which automate part of the translation process, text editors, which ease the "manual revision" part of conversion, and programs which can scan another program and print a flowchart for it.

---

1/See app. I, pp. 21 and 22.

Conversion costs can also be reduced by management recognition that most production programs will eventually be converted to newer equipment. Therefore, it is prudent to make every effort to avoid the use of vendor-unique features present in programming languages. Such avoidance will aid in reducing the conversion costs involved in new equipment procurement, and facilitate maximum competition.



## CHAPTER 3

### CONCLUSIONS, RECOMMENDATIONS, AND AGENCY COMMENTS

#### CONCLUSIONS

Software is the most costly element of modern ADP systems. It is both the highest single cost of developing an automated system and the system component with the most potential for adverse impact on the user tasks it supports. Malfunctions in these tasks have a potential for cost impact greater than the cost of developing the software.

The conversion of software to replacement hardware systems is a recurring, frequent, and costly activity in the Federal Government. Program conversions among installations are now much less frequent, but greater sharing of software would increase them. Reduction of the time and effort needed for this type of conversion would encourage software sharing.

More effort devoted today to developing better quality software and documentation would ease eventual conversion problems. Means of attaining better quality include better enforcement of standards (for programming languages, programming practices, and documentation), adoption of the new programming technologies, better training for programmers and systems analysts, and adoption of productivity aids.

The greater cost of developing higher quality software would be offset by (1) easier and cheaper future conversion of the software, (2) fewer errors in the supported user tasks, and (3) the increased feasibility of sharing the software by multiple users. Also, the savings from cooperative efforts to develop software for groups of users, and the productivity improvements possible with new programming technologies, mean that developing better software need not necessarily cost more than the old methods.

More widespread use of programmer productivity aids would also ease both software conversion and original software development problems. These aids include automatic translation programs and programs which automatically scan another program and print a flowchart for it.

Management, anticipating eventual conversion, can direct that vendor-unique features be avoided where possible 1/ in developing new applications for existing equipment. By avoiding such use, eventual conversion to replacement equipment of other manufacturers will be eased.

We estimate that in today's environment \$100 million can be saved annually through the use of techniques and activities cited above. We believe that achievement of these savings could be facilitated through establishing a Federal center for software conversion which would provide a pool of expertise for agency assistance.

In the long run, even more savings in the software area may be attained as a result of other actions, especially standardization of more high-level languages and further improvements in programing and documentation techniques. We plan to monitor these developments and report on them as appropriate.

#### RECOMMENDATIONS

We recommend that OMB assist in establishing a center of Federal expertise for software conversion. The services provided by this center should include impartial estimates of the conversion implications of system procurement alternatives, contracting for software conversion, some capability to actually perform conversions, and a mechanism for transfer of lessons learned to future software conversions, developments, and procurements. GSA should be able to require independent conversion estimates by the center for expensive agency procurements which have significant conversion costs.

---

1/In cases where vendor-unique features must be used, we believe that management should require (1) that such use be justified by savings in operating costs, (2) that the justification be documented, and (3) that the use of the unique features be isolated into separable parts of the program (called modules).

We also recommend that the heads of Federal agencies emphasize quality in software development, promote standards more vigorously, and require both their in-house staff and their contractors to observe standards. Standards to be promoted include programing languages, documentation, and programing practices. The guidelines recently published by NBS (FIPS PUB 38) 1/ offer a reference for documentation. Programing practices standards should include restriction of the use of vendor-unique features in software development. 2/

We recommend that NBS select and publish a standard set of programing productivity aids for Government-wide use. These aids would ease both the original development of software and its later maintenance and conversion. Here, "standard set" means standard in the sense that each aid's features and uses would appear the same to programers no matter which make of computer it was installed on.

#### AGENCY COMMENTS

We asked HEW, DOD, Commerce (NBS), OMB, and GSA to comment on our preliminary report. Their replies, which indicated general agreement, are shown in appendix III and discussed below.

The Inspector General, HEW, said that he had no substantive comments.

The Deputy Assistant Secretary of Defense (Comptroller) said that he agreed that sufficient benefit would accrue to make implementing our recommendations worthwhile, even though he was "less optimistic about the potential for hard dollar savings" than we are.

---

1/Federal information processing standards publication 38.

2/Another possible means of restricting the use of vendor-unique features is that of requiring vendors to sell the Government programing language compilers which include only the standard language (no special features unique to the vendor). This would force programers to avoid special features by removing them completely.

The Assistant Secretary for Administration, Commerce, concurred with our three recommendations. She stated that NBS recognizes the need for greater use of productivity aids and will work in this area as priorities dictate and resources allow.

The Associate Director for Administrative Management, OMB, commended the report and agreed to support the establishment of a conversion center. He raised the question of whether or not the center should be the only source of conversion services Federal agencies would be allowed to use. He suggested that agencies be allowed to use other sources as a way of subjecting the proposed center to competitive pressure. He also indicated that he feels the Federal Computer Performance Evaluation and Simulation Center is an example of how to arrange the conversion center.

In general, we agree that an arrangement like the above center has (i.e., agencies voluntarily requesting assistance) is appropriate for the conversion center. However, we believe GSA should be able to require an independent estimate by the conversion center for expensive agency procurements in which conversion is a significant component of the total cost. We modified the report to clarify this matter.

The Administrator of GSA supported our conclusions and recommendations, but said that we should have discussed conversion in terms of the broader problem of management planning for future needs. We also realize that conversion occurs in a broader context of related matters, but feel that our report covers a subject which warrants attention and action. He said that while adoption of our recommendations would be beneficial, it would not substitute for good ADP installation management. We agree.

The Administrator discussed some software conversion problems, including mixes of agency-developed and contractor-developed software, and "efficient nonstandard programs." We realize that many Federal ADP installations have software developed by outside sources as well as by their own programmers. We realize also that data base management systems are written in assembly languages (nonstandard) in the current state of the art and would not normally be converted.

Concerning the use of unique features of programming languages--which we advise against--the Administrator said that these features often provide advantages over the near term. We do mention that they must sometimes be used. However, the experts we consulted during our review felt that the near-term advantages of unique programming features are often

overrated and contribute unnecessarily to increased long-term costs without really reducing short-term costs.

The Administrator disagreed with our suggestions to use industry-provided application software, saying that its lowering costs and speeding implementation are the exception, not the rule. During our review, we found that many users of purchased software are satisfied and intend to buy more. Also, user groups--including those of the International Business Machines Corporation and the Control Data Corporation--repeatedly discuss purchased software as a viable and commonly used alternative to software written by the organization's own programmers. It is true that such software must often be modified, but modification can be much cheaper than writing complete new software.

The Administrator said that it would have been helpful if we had more fully discussed the involvement of programming language standards with conversion. We are aware of the important effect of standards (or lack of them) on conversion. We agree that better standards, together with compliance mechanisms, would greatly reduce the problem. We are now studying the Federal Information Processing Standards Program and will report our results to the Congress.

WHERE COMPUTER SOFTWARE IS NOW AND HOW IT GOT THERE

In 1951, the Census Bureau installed the first business type computer in the Federal Government. The computer itself was expensive, large, and delicate. Its programming was done in machine language by a problem solver untrained in programming. The cost of this software was small compared to the cost of the hardware it controlled.

From that beginning, software has evolved until it is now generally the most costly single part of an ADP system; programming is recognized as a separate occupation which can be taught and managed; and there exists a very large investment in programs running on current computers, many of which will eventually be converted to run on other computers.

Several characteristics and issues in the software area were identified repeatedly in the literature and by practitioners.

IMPORTANCE OF APPLICATIONS SOFTWARE

An industry group concerned with the future of ADP recently predicted that by 1985 software will be about 90 percent of the total cost of a computer system. Also a recent GAO report, "Improvements Needed in Managing Automated Decisionmaking by Computers Throughout the Federal Government" (FGMSD-76-5, April 23, 1976), found that "software problems" are a major cause of expensive malfunctions of automated systems.

Many administrative applications are automated in virtually every agency of the Government, and it is applications software that drives this automation. The Federal Government alone spends about \$6 billion a year on computer software and now has large inventories of computer programs, many of which will eventually be converted to run on another computer system. This annual cost includes software conversion projects which are frequently expensive. For example, a single Navy conversion project reported approximately

--100 staff-years of effort,

--a total direct labor cost of about  
\$1.2 million, and

--conversion of 400 applications programs over  
a 12-month period.

Software conversion costs and difficulties are factors in many procurements of new computer systems. The difficulty of conversion is often cited as a reason to buy a replacement computer from the same vendor.

#### WHY SOFTWARE HAS BECOME SO IMPORTANT

Two causes have made software a relatively larger part of ADP systems costs: one is increasing complexity and quantity of software functions demanded by users and the other is decreasing hardware cost.

The first is due to the increased use of computers by laymen who want convenient automation which is tailored to (1) their problem at hand and (2) their convenience--not the convenience of the machine or its programmers. The new user-oriented applications are very different from older clerical applications, such as payroll, which were the original uses of business automation. Programs which provide user convenience are more complex, and since users demand convenience, the programmers' work is becoming more complex.

Besides more complex software, there is simply a demand for more software; there are more computers, and they all need software to make them run. With less expensive computers, more organizations find more automation attractive, and again more software is needed. The growth in use of computers is shown by the following table.

	<u>Computers installed</u>		
	<u>1965</u>	<u>1970</u>	<u>1975</u>
U.S. Government	2,412	5,277	8,649
Nationwide	23,000	70,000	175,000
	<u>Programers employed</u>		
Nationwide	80,000	175,000	215,000

The steady decline in the cost of pocket calculators illustrates the second cause (cheaper hardware). The calculators use technology similar to that of computers, which are also going through similar decreases in cost per task done. The decreasing cost of electronic computing hardware makes software a proportionately larger part of the total costs of ADP systems.

TRADITIONAL PRODUCTION AND USE  
OF APPLICATIONS SOFTWARESoftware development

The life cycle of a unit of computer software is divided into development time and operational time. Development time ends when the software is in production automation of the user's task. Traditionally, a software development project has had these characteristics:

- It cost more than was expected and ran late. Adding more people to late projects often did not restore them to schedule.
- The first production version delivered was really a prototype in conventional engineering terms. Besides the software itself being a prototype, its documentation was often sketchy or missing.
- Because a prototype was delivered, the postdevelopment costs of fixing and modifying have typically been as great as the development cost and sometimes far greater.

Several factors contributed to the situation. First, the invisible nature of both the work process and its product made software projects very difficult to manage and predict. Second, the explosive growth of the use of computers created a great demand for programmers which, in turn, demanded the use of many new programmers, most of whom learned on the job. Frequently, low productivity and poor quality of product resulted. Third, there was little idea then of how to train programmers properly. Many were largely self-taught. Fourth, a tradition grew of programmers as secretive craftsmen whose products during development, were their own property. Fifth, little attention was given to reducing some of the clerical tasks of programming. Programmers' productivity was low because much of their time was occupied with such things as key-punching and running errands rather than programming.

Software operation

The operational part of the life cycle lasts from the beginning of production with the first version to the time that the software is replaced or discarded.



During this period, costs other than normal processing ones may be incurred to correct errors in the software, to modify it to add new functions, or to convert it to run on another computer. They are all postdevelopment costs.

Corrections of errors and modifications are commonly combined under the term "maintenance." The percent of programmer time spent on maintaining software developed in the traditional way has been estimated from 20 percent to 80 percent.

Software conversion costs are those special post-development costs which are incurred to make the software run on some other computer than the one for which it was written. They include delays in users' tasks, retraining, programming labor, interference with other uses of the computer, documentation revision, and possibly lowered employee morale.

We feel that 50 percent of total software costs is a representative value for all postdevelopment software costs, including conversion.

#### LARGE GOVERNMENT INVESTMENTS IN ADP

Federal investments in ADP to date include:

- Computer equipment. The table on page 16 shows the large number of computers now in place. GSA's reported fiscal year 1975 Federal ADP inventory shows almost \$3 billion in owned hardware and over \$1 billion in leased hardware. These computers, and their software, range from early to recent models.
- Programs. There are now large numbers of computer application programs in Federal agencies. Tremendous effort has been spent on developing these programs; documenting, maintaining, and modifying them; and training operators and users. While the total investment cannot be estimated with any accuracy, its magnitude is indicated by the inventory of programs at the National Institutes of Health Computer Center in Bethesda, Maryland. At a recent meeting, NIH representatives indicated that this single installation has about 40,000 programs, with an estimated 10 million lines of code.

Using an original development cost of \$8 per line of code would yield a software development cost over the years of about \$80 million for just this one installation. While this figure is only an estimate, the inventory certainly represents a large expenditure.

--Training. GSA reported that 33,694 Federal staff-years were spent on systems analysis and programming in fiscal year 1975. The cost of training that number of persons in the current methods of developing software, programming languages, and knowledge of specific programs is difficult to estimate but is certainly very large.

### A SOFTWARE INDUSTRY NOW EXISTS

In the early days of computers, almost all applications software was developed by programmers who were employed by organizations operating the computers. Now, a separate industry produces applications software to sell.

At a recent industry conference, the publisher of a software journal pointed out that total software industry sales have grown from \$4.5 million in 1970 to \$750 million in 1975, that there are now 400 firms that sell software, and that about 3,500 different software packages are offered by these firms; of these, 2,200 are application oriented. A strong point in favor of using software that is already developed is that it is much less expensive to convert it to the new user's computer, if that is necessary, than to develop similar software anew. In some cases, new development can cost 10 times as many dollars and take 10 times as long. Many common tasks have been automated with programs generalized so as to be useful to many organizations.

This software industry is an alternative to Government agencies' producing their own applications software; use of this alternative could lower costs and speed implementation dates.

### SOFTWARE PROBLEMS GENERALLY

Historically, projects to develop original software as well as conversion projects have been completed later than scheduled or at a higher cost than predicted, or both. Many causes contribute to this situation--some

managerial, some sociological, and some technical, as discussed below.

### Managerial

The software field is still relatively new and lacks accurate means both in measuring its productivity and predicting its production. Workers who produce software practice an ill-defined and difficult-to-quantify craft. They are commonly considered to be in demand and difficult to retain and to manage effectively. Customers (end users) for whom the software is created often do not have a good definition of the process or function that they want automated. Changes requested after projects have started, which seem trivial to the customers, have often caused major rework efforts with consequent delays and increased costs.

### Sociological

The experts and literature that we consulted indicated that many computer programmers view themselves as craftsmen, with strong feelings of "I'd rather do it myself." This attitude was echoed by the programmers who responded to our questionnaire. Traditionally, the programmers' attitude was indulged because they were scarce. There is a feeling that if a program written elsewhere is brought into an installation then the local people must not be competent. This feeling has frequently caused installations to duplicate programming that had already been done elsewhere.

### Technical

Programs are often designed and written hastily to meet deadlines and tested and documented inadequately or not at all. Thus, quality is sacrificed for urgency. Programs have been typically devised to run on one specific computer for the local task(s) of its owners. Little or no consideration is given to the probable eventual conversion of the programs.

Each manufacturer supplies aids to programmers which include features peculiar to that manufacturer's equipment. When programs written with these tools are to be converted to another manufacturer's equipment, these peculiarities must be replaced because they no longer work. This causes problems and increases costs.

Federal ADP procurement methods, intended to insure competition among prospective hardware suppliers, make it very possible that an installation's replacement computer will be purchased from a different manufacturer than the one the present computer was purchased from. For a variety of reasons, it is more difficult to convert programs running on one make of equipment to another make than to convert them to a later model of the same make. Finally, mechanisms for transfer of technical knowledge about conversions from one Federal agency to another are primitive.

Another problem with both technical and sociological aspects is documentation, which is material prepared to explain a computer program. Documentation is human-readable material over and above the actual code which drives the computer. Since the programmer usually has a requirement to make the program work as soon as possible, documentation is often deferred until after the program is running. Once a given program is running, a new programming task may cause the work of documenting the first program to be shunted aside. Also, programmers tend to shirk documentation, because it is less creative than actual programming. When that first program is later modified or converted, the work is usually done by someone other than the originator. When documentation is missing, incomplete, or obsolete, the program conversion effort often must duplicate a great deal of the original development work.

#### NEW SOFTWARE TECHNOLOGIES

The recognition of a software crisis in the mid-1960s caused considerable attention to the need for improving both the management and the technology of computer software. Several organizations undertook projects to devise better methods.

Their efforts have produced a group of software innovations, collectively referred to as the new software technologies, which include the following concepts:

- Software engineering: the idea that an engineering approach can be used in developing software, as it has been in developing hardware. Software engineering includes many methods for the controlled design and development of high-quality software, including

both management approaches, such as chief programmer teams, and technical approaches, such as structured programming.

- Chief programmer team: a small team consisting, as a minimum, of a very superior programmer as a leader, a backup programmer, and a programming librarian or programming secretary. The team is supported by an automated facility called a development support library.
- Structured programming: a way of arranging the actual code of computer programs (by their authors) so that they will be more easily understood by others who must later maintain and modify them.
- Development support library: an automated facility which is used to maintain software development files, including programs, data, and documentation. The library is maintained, according to a standard set of procedures, by a trained person who is sometimes called a programming librarian or programming secretary.

A number of organizations have reported significant improvements in terms of productivity, ease of maintenance, and reduced errors in software development by adopting these techniques.

#### STANDARDIZATION EFFORTS

There have been a number of efforts to standardize facets of ADP, including software. A unique effort was the adoption of the American National Standards Institute 1968 standard for the Common Business Oriented Language (COBOL) as a Federal Information Processing Standard. As of late 1977, COBOL is the only programming language for which a Federal standard exists. Attempts to comply with this standard have reduced the differences between different vendors' versions of COBOL. Meanwhile, technology has advanced. A 1974 COBOL standard has been adopted which is considerably different from the 1968 version. Some existing COBOL programs require modification before they will work with the 1974 version of COBOL. This modification is also a type of conversion.

ORGANIZATIONS' VESTED INTERESTS  
IN EXISTING METHODS

The vested interests of users, data processing entities, standards groups, and management hierarchies are other factors in the software area. Organizations have adopted ADP, with its attendant investment in goods and people, as a part of their structures and a component of their budgets. Other organizations have been established to regulate the users. Both the adoption and the establishment were long, slow processes, and considerable inertia may now exist against sweeping changes.

CURRENT EFFORTS TO ADDRESS SOFTWARE  
CONVERSION ON A GOVERNMENT-WIDE BASIS

Some efforts currently address software conversion on a Government-wide basis. They include work by GSA, the Navy, the Air Force, and the House Committee on Government Operations. Each is discussed below.

In October 1975, GSA received a contractor's study entitled "Feasibility Study For Development of a Program Toward Competitive Procurement of Higher-Order ADPE." The relevant findings are paraphrased below:

- The movement of a particular ADP system workload to another ADP system (conversion) appears to be the major impediment to initiating full and open competition in a procurement to replace or upgrade ADP because of its costs. The cost of conversion is controlled by a number of factors, including inadequate or improper documentation, personnel not available to complete or correct documentation, programs written for specific hardware characteristics, programs not written in higher level language 1/, and programs in higher level language using nonstandard or machine-dependent programming techniques.
- While there is basic agreement among ADP vendors and users alike that conversion problems are the main stumbling block to competition, there is little

---

1/Higher level languages provide a programmer with a more convenient means of writing a program which is then translated automatically into a code that the machine can follow directly.

information available on the subject. Conversion experience is either not transferable or closely held. Hardware vendors do have considerable documentation and experience in conversion but tend to treat it as proprietary.

--Effective management is the key to minimizing conversion costs. Since system life is in the range of 5 to 10 years, it is imperative that ADP management plan for system replacement.

--The first basic management tool to be implemented should be effective program library controls and reporting procedures. Any replacement upgrade recommendation logically must be preceded by a study which determines that higher capability is needed. Such a study must include, as key elements, the condition, and efficiency on existing equipment, of the application programs. Proper program library procedures would contribute significantly to the study by supporting statements about the number of programs to be converted, locating similar converted programs in other libraries, and showing whether or not programming standards are followed.

--The next management tool that should be implemented is a standard for machine-independent programming [languages] and systems design to prevent equipment lock-in to single vendors. Both history and the anticipation of the future indicate that machine-independent software is cheaper in the long run.

--A third management tool is the elimination of interim ADP equipment upgrades. These indicate failures of ADP management. Private-sector services could be used for workload that exceeds the present system while a replacement is sought.

--A fourth tool is the charging of conversion costs to ADP operations instead of to the procurement process. ADP management is responsible for the status of the software, and the status of that software, including adequate documentation and higher level language, affects operations today as well as in the future. Charging conversion costs to procurement hides ADP management problems and virtually eliminates competitive pro-

curements. [GAO comment: This viewpoint is in opposition to many ADP managers, who claim that conversion cost should be considered part of the cost of each procurement alternative.]

The Information Systems Division, Office of the Chief of Naval Operations, studied the costs required to convert its installations' operations into regional data processing service centers. The size of this effort (13,000 programs totaling about 8 million lines of code and associated files) and the presence of software conversion as a recurring Navy problem led to a feasibility study of a centralized conversion team in the Navy. The report, entitled "Feasibility and Desirability of Establishing a Centralized Conversion Team Within the Navy," recommended that a centralized Navy conversion staff be established. It was to provide Navy installations with

- workload analysis and planning,
- general consulting,
- training in conversion techniques,
- "hands on" conversion performance, and
- contractor selection and contract management for conversions done by contractors.

The study group was unable to quantify the savings that a centralized staff would effect but stated that they were certain that savings in money and staff time would result.

At GSA's request the Navy's Federal COBOL Compiler Testing Service <sup>1/</sup> did a "Feasibility Study of a Federal Data Processing Service Center for Computer Systems Conversion Support" which suggested a Federal entity for software conversion to provide two categories of service to client agencies--analysis and solicitation. A cooperative venture with GSA was suggested similar to, but separate from, the Federal COBOL Compiler Testing Service.

---

<sup>1/</sup>The Federal COBOL Compiler Testing Service tests vendors' versions of COBOL to see if they comply with the Federal Standard.



In December 1976, GSA representatives indicated that an Interagency Agreement and Federal Property Management Regulation were planned.

The U.S. Air Force made a study of software conversion for its fiscal year 1976 Worldwide ADP Single Managers' Conference. The study, entitled "ADP Conversion Cost," recommended that:

- Conversion costs which can be clearly and fully satisfied and documented be included in the overall costs that are to be considered in the selection of equipment according to GSA's Federal Management Circular 74-5.
- Air Force Regulation 300-2, paragraph 4g, be clarified by the addition of the words "and conversion" to read: "Total life cycle costs (not to exceed 8 years), including costs of acquisition and conversion, must be considered in determining whether the procurement should be sole-source or competitive." [GAO comment: This conflicts with the finding of the GSA contractor study cited above.]
- Conversion cost tracking be added to Air Force ADP Project management to accumulate data to improve future conversion estimates.
- The question of software system redesign be separated from that of software conversion.

In June and July 1976, the House Committee on Government Operations, headed by Congressman Jack Brooks, held hearings on Federal ADP operations. On October 1, 1976, the Committee issued a report entitled "Administration of Public Law 89-306, Procurement of ADP Resources by the Federal Government."

The hearings and the report surfaced conversion to replacement systems as a major concern in the management of Federal ADP.

DETAILS OF THE SEPARATE SOURCES OF INFORMATION

This appendix presents details from our sources of information--experts and literature, and the questionnaire data collection. While our sources substantially agreed about causes of problems and remedies, and these are shown in the text, there were minor differences of detail and emphasis. These slight differences reflected the different perspectives and orientations of the sources. Substantial agreement emerged on the major points, however.

EXPERTS AND LITERATURECauses of problems

Experts and literature identified the following as major causes of software conversion problems:

- Lack of readily available software conversion expertise. Such expertise is needed and concepts of it ranged from that of consulting, which would estimate conversion costs to guide procurements, to that of an entity which would actually perform conversions for agencies. A contractor-operated entity to perform conversions was also suggested.
- Poor planning of conversions. Unreasonable demands of users and failures of programming management to estimate resources correctly, to schedule realistically, and to assign trained people to conversion projects were all cited as consequences of poor planning. Another planning difficulty cited was that of allowing conversions to be complicated by demands for modifications to the software being converted.
- Procurement policy. Several experts criticized the Federal low-bid hardware procurement policy, saying that ignoring conversion aspects of a procurement decision heavily penalizes organizations that must convert their applications to another manufacturer's computer. (Our comment: We believe that, if software development anticipated eventual conversion and took measures such as sharply reducing the use of vendor-unique

features by programers, this problem would be much less important.)

- Vendor-unique features. These features are plentiful in older applications software. They are there because of a lack of standardization among vendors, and include vendor-unique features of programming languages (called extensions), dependence on unique hardware, etc., and were used to make original development easier or to make programs use fewer computer resources. Modifying a program that contains vendor dependencies to remove them may be easy or it may be so difficult that writing a completely new program is easier than converting the old one. However, the original writing of the program can often be planned to avoid such dependencies.
- Poor quality of the material to be converted. This category included poor quality of (1) the computer programs themselves and (2) their documentation. Poor quality of the programs can include errors and omissions, inefficiency, and incomprehensibility to anyone but the author. Poor quality of documentation may mean that it (1) does not exist, (2) is incomplete, or (3) is obsolete. (Obsolete documentation means that changes to the documentation have not kept up with changes to the software.) Perhaps the worst case is that of a currently dated documentation that was not brought up to date before it was reprinted. It looks current, but it is not.
- Absence of productivity-aid facilities. Facilities identified as missing and needed in many conversion situations include interactive terminals, automatic translation programs, and aids for the manipulation of source code. The first aid provides programers with quicker response to tests of the converted programs, and the other two automate some of the work of changing the program.
- Lack of people trained in conversion. There seems to be no effective mechanism today for transmitting conversion experience from one project to another. We know of no formal training in conversion available within the Government. Frequently, junior people are

assigned to conversion projects. One reason is that conversion is deemed a less prestigious task; therefore, senior programmers avoid it. On the other hand, the private sector has several firms which specialize in conversion services.

- Confusion of conversion effort by concurrent redesign. Sometimes the task of converting software becomes entangled with augmenting or redesigning it. Conversion means making the software produce the same outputs on the new computer (from the same set of inputs) as on the old one. Augmentation means to add functions to the program. Redesign may add functions or achieve the old functions in a different manner. The often difficult task of conversion is frequently further complicated by concurrent user requests for augmentation or redesign.

#### Suggestions for improvement by experts and literature

Experts' recommendations for conversion improvements varied, depending upon their backgrounds. Vendors' ideas included using their own products or services; persons in standards organizations recommended greater emphasis on standards; and a member of a DOD central software design agency recommended further centralization of software development. However, agreement emerged on some points, which are discussed below.

- Federal center for conversion. Several experts feel that a Federal center for software conversion would be extremely valuable. They offered various concepts, from that of a small group which would estimate conversion costs for specific procurements to that of a larger entity. Both would estimate and actually perform the below-listed services for client agencies. Concepts of its organization ranged from that of a completely Federal staff and facility, through a Government-owned contractor-operated facility, accompanied by an interagency agreement on management. Such a Federal center for conversion could:

1. Assess conversion aspects of specific procurements or upgrades by client agencies.

2. Contract for software conversion services for client agencies.
3. Cut conversion costs by performing conversions for less skilled agencies.
4. Train in-house conversion teams from client agencies.
5. Help client agencies evaluate the suitability to their needs of using other software than that proposed for conversion.

--System procurement alternatives. Some of the experts felt that conversion costs should be given more consideration in the selection of large systems because (1) the high cost of conversion to a new vendor may cancel any hardware savings and (2) conversion difficulty may cause a long, expensive delay in user functions. Such a consideration of conversion costs and time implies that some impartial entity should estimate them. This entity should have extensive experience with large-scale conversions.

--Human factors. Most of the experts identified these as crucial to the success of both software development and software conversion. Selection, retention, and continued training of analysts and programmers are felt to be deficient. The large variation in ability to create software should be recognized, and better means of identifying and training good performers should be developed. Modern productivity aids should then be used to support the good performers.

--Software development that anticipates conversion. Software development project managers can anticipate eventual conversions and take steps to reduce problems with them. Such steps can include standards for programming practices, such as limiting programmers' use of vendor-unique features, and requiring better documentation.

--Further centralization of original software design and development. This offers a large potential for savings through eliminating redundant software development and the ability of the central developer to concentrate resources, the cost of which

will be served by multiple installations. Such centralization could reduce conversion costs--a better job could be done of design and implementation because multiple users would share its increased cost. Such centralization may imply displacing programmers from field installations.

Management and organization must recognize and support technical methods before their potential will be realized. <sup>1/</sup> While technical methods identified by experts and in literature include:

- Government-wide distribution of automated conversion aids and conversion manuals. Such programs and other materials could be selected, for the COBOL and FORTRAN <sup>1/</sup> languages, from items already developed. Management emphasis should accompany their dissemination.
- Standards for programming practices. These should be forced for new software development, for both Federal and contractor-developed software. (Some agencies are already active here.) These standards could include the new programming technologies for software creation and some restrictions on the use and documentation of programming language features peculiar to a particular hardware vendor.
- Programmer productivity aids. These should be installed and their use encouraged wherever possible. They include programs which print flowcharts of other programs and interactive terminals.
- Training. Systems analysts, programmers, and managers should be trained in the new programming technologies. Management should encourage their use in both software creation and conversion.

---

<sup>1/</sup>Formula Translator.

THE QUESTIONNAIRE DATA COLLECTIONAdministration

The questionnaire data collection was done to assess:

- Working-level perceptions of the causes of software conversion problems, and the value of various means of improvement.
- The importance and frequency of conversion--how often programmers are involved in it, etc.
- Data on some related matters, including programmer demographics, language usage, training available, and programmer activity breakdown.

The mechanics of administration emphasized respondent anonymity as much as possible to get honest input from the working level. The questionnaire was administered in a classroom situation by our representatives. The respondents were briefed, and the questionnaires were collected immediately after completion. Thus, local hierarchies had no way of knowing what a given individual wrote on a questionnaire.

Two quite similar questionnaires were administered--one to programmers and the other to their supervisors. At a given installation, the supervisor group and the programmer group were separated during the completion of their respective questionnaires. To give the respondents maximum flexibility, most questions were constructed with a blank "other" choice where respondents could write in their own answers, as well as a convenient choice of answers.

Our representatives who administered the questionnaires reported that:

- There was no evidence of special selection of respondents by their local hierarchies.
- There was no evidence of prebriefing of the respondents on the subject.
- Respondents did not see copies of the questionnaire ahead of time.
- In almost all cases, respondents were favorably disposed toward the effort and indicated some interest in the subject.

--In some cases, respondents expressed considerable interest in the subject, including desire to see the results.

Processing of the questionnaires further confirmed the respondents' interest in and good motivation toward the subject:

--Almost all of the collected questionnaires were completed carefully enough to be useful (1,983 out of 2,005 programmer questionnaires and 458 out of 465 manager questionnaires).

--The last question on each questionnaire (left completely blank for respondents' other discussion) was written in by over 21 percent of the programmers and over 33 percent of the supervisors (after they had completed over 30 other questions). Also, there were numerous write-ins to the blank choices made available in the preceding questions.

--A number of caustic write-in answers, such as "This questionnaire is a waste of time," showed that the respondents believed that they would be anonymous.

#### Description of sites and respondents

The data collection reached 43 different ADP installations, 2,005 programmers, and 465 supervisors. The sites ranged from the smallest one, where 2 programmer and 2 supervisor questionnaires were collected, to the largest one, where 325 programmer and 91 manager questionnaires were collected. Figures 1 through 6 describe the sites and respondents.

Figure 1

<u>Region</u>	<u>Site visited</u>		<u>Total</u>
	<u>DOD</u>	<u>Non-DOD</u>	
Kansas City	16	10	26
San Francisco	3	6	9
Detroit	<u>7</u>	<u>1</u>	<u>8</u>
Total	<u>26</u>	<u>17</u>	<u>43</u>



Figure 2: Site populations by region:

<u>Region</u>	<u>Average programers per site</u>	<u>Average supervisors per site</u>
Kansas City	60	12
San Francisco	39	11
Detroit	13	7

Figure 3: Experience of respondents--years  
in data processing:

	<u>Mean</u>	<u>Median</u>	<u>Mode</u>	<u>Number with over 3 years</u>
Programers	9.2	8.1	8.0	a/1,565 out of 1,976
Supervisors	12.6	13.2	15.0	a/ 398 out of 458

Figure 4: Experience of respondents--years  
in present type of job:

	<u>Mean</u>	<u>Median</u>	<u>Mode</u>	<u>Number with over 3 years</u>
Programers	5.1	4.1	1.0	a/1,100 out of 1,962
Supervisors	5.3	3.9	1.0	a/ 242 out of 457

Figure 5: Programers' and systems analysts'  
categorization of their present jobs:

<u>Category</u>	<u>Number</u>	<u>Percent</u>
Definite business orientation	645	32.5
Definite scientific orientation	312	15.8
All other answers, such as "senior analyst," "coder," etc.	<u>1,026</u>	<u>51.7</u>
Total	<u>a/1,983</u>	<u>100.0</u>

a/The numbers of respondents differ in figures 3, 4, and 5 because slightly different numbers of people answered the questions.

Figure 6: Programers and systems analysts' experience with various programing languages:

<u>Language</u>	<u>Extensive work experience</u>	<u>Moderate work experience</u>	<u>No work experience</u>
	(percent)	(percent)	(percent)
COBOL	59.3	18.8	21.7
Assembly	38.0	24.5	37.4
FORTTRAN	28.0	17.9	54.0
Machine language	14.4	32.8	52.8
Data management languages	6.6	11.1	82.3
JOVIAL	6.2	2.1	91.7
BASIC	6.0	11.5	82.4
PL/I	3.2	5.7	91.0
RPG	2.7	12.4	84.9
ALGOL	1.9	2.8	95.2
GPSS	0.4	3.1	96.6
SIMSCRIPT	0.1	0.9	99.1

#### Causes of software conversion problems

Both groups evaluated two lists of causes of conversion problems--one list for conversion to a replacement system and the other for conversion of foreign programs. Both groups' choices on the lists were ranked by counting those who answered that a choice was "at least moderately important." Figures 7 through 10 show supervisors' major causes of difficulty for replacement systems, programers' major causes for replacement systems, supervisors' major causes of difficulty of converting foreign programs, and programers' major causes for converting foreign programs, respectively.

Figure 7 shows that the supervisors ranked poor documentation of the old application programs as the most important cause of difficulty of conversion to replacement systems. A close second was the inadequate weighting of conversion considerations in selecting the replacement system to which the software had to be converted. The five highest ranked causes shown can be regrouped into two broad categories: (1) poor quality of the material to be converted, which includes the first, third, and fifth causes and (2) poor choice of the replacement system, which includes the second and fourth causes.

---

Figure 7: Supervisors' major causes of problems with conversion to replacement systems:

<u>Percent ranking cause important</u>	<u>Cause of problems</u>
87.4	Inadequate or missing documentation of application programs.
87.1	Conversion considerations are not adequately weighted in the system selection process.
83.1	Many application programs are monolithic and "patched;" that is, they are NOT well structured, modular, or well organized.
82.0	Policy of selection of low bidder could force us to accept a less compatible system, such as one made by a different manufacturer.
79.2	Combination of personnel turnover and poor documentation has made many application programs virtually unintelligible.

---

Figure 8 shows that the programmers selected the same five causes as most important, with the same two ranked first and second, as the supervisors. The third, fourth, and fifth choices are in different order, but the differences are slight.

---

Figure 8: Programers' major causes of problems with conversion to replacement systems:

<u>Percent ranking cause important</u>	<u>Cause of problems</u>
80.3	Inadequate or missing documentation of application programs.
77.4	Conversion considerations are not adequately weighted in the system selection process.

- 73.4 Policy of selection of low bidder could force us to accept a less compatible system, such as one made by a different manufacturer.
- 73.2 Combination of personnel turnover and poor documentation has made many application programs virtually unintelligible.
- 73.1 Many application programs are monolithic and "patched"; that is, they are NOT well structured, modular, or well organized.

Figure 9 shows that the supervisors rank the need to add new features to foreign programs as the highest cause of difficulty with their conversion. These five causes can be regrouped into three broad categories:

- Selection of unsuitable foreign software for conversion (first and fifth causes).
- Poor quality of the material to be converted (second and third).
- Differences between vendors' products (fourth).

Figure 9: Supervisors' major causes of problems with conversion of foreign programs: (note a)

<u>Percent ranking cause important</u>	<u>Cause of problems</u>
75.7	The task of conversion is complicated by the frequent need to add new features to the programs being converted.
73.7	Foreign programs are often poorly or inadequately documented.
72.5	Foreign source code is often poorly organized or does not contain embedded comments/notes/remarks; therefore, it is hard to read and comprehend.

a/A foreign program was defined on the questionnaire as a program brought in from another installation.

- 67.0 Differences in the features readily available with different vendors' operating systems affect the conversion of applications programs.
- 62.9 Foreign programs need too much modification to the needs of another installation to be worth it.

Figure 10 shows that the programmers ranked their own lack of knowledge highest, with documentation a close second. The programmers' five causes can be regrouped into four broad categories:

- Lack of knowledge of conversion (first choice).
- Poor quality of the material to be converted (second and fourth choices).
- Poor management (third choice).
- Unsuitable material to be converted (fifth choice).

We believe that the slightly different opinions of the programmers reflect the fact that they and the supervisors have a somewhat different perspective on conversion projects. (The supervisors ranked the "lack of knowledge" choice at 56.0, showing that they consider it considerably less important than any of their own top choices.)

Figure 10: Programers' major causes of problems with conversion of foreign programs:

<u>Percent ranking cause important</u>	<u>Cause of problems</u>
76.8	People assigned to conversion projects often have little knowledge of what they might encounter.
76.5	Foreign programs are often poorly or inadequately documented.

- 72.7 Programers' supervisors allow themselves to be talked into unreasonable deadlines.
- 70.7 Foreign source code is often poorly organized or does not contain embedded comments/notes/remarks; therefore, it is hard to read and comprehend.
- 70.2 The task of conversion is complicated by the frequent need to add new features to the programs being converted.

---

Figure 11 shows that the supervisors ranked better training for programers as the most valuable source of improvement of conversion to replacement systems. The five means can be regrouped into three.

- Improvement of the performance of programers doing the work (first, third, and fourth choices).
- More consideration of conversion in the selection of replacement systems (the second).
- Improvement of management (the fifth).

The strong ranking given to the three choices in the first category indicates that the supervisors feel that programer training would be a very valuable means of improvement.

---

Figure 11: Supervisors' major means of improving conversion to replacement systems:

<u>Percent ranking method valuable</u>	<u>Method of improvement</u>
88.7	Better training for programers in superior programing practices.
83.7	More weight to conversion cost and problems in the selection of replacement systems.
82.7	Better training for programers in conversion problems and techniques.

- 82.7 Ability to send the conversion team to appropriate specific training before starting work.
- 81.7 Better training for installation managers in conversion planning and estimating.

Figure 12 shows that the programmers also value training highly, for themselves (highest), for their supervisors, and for user management. These five choices can be combined into (1) training for programmers (first, second, and fourth) and (2) training for management (third and fifth). This result indicates that the programmers attach even more importance to training than their supervisors do.

We believe that the slight difference seen between the programmer and supervisor evaluations is due to their different perspectives. The managers' probable involvement with the selection decision gives them a broader view.

The emphasis on training here, coupled with the importance of the poor quality cause, indicates that training programmers to do a better job of originally writing programs is valued highly as a way to reduce future trouble with converting those programs.

Figure 12: Programers' major means of improving conversion to replacement systems:

<u>Percent ranking method valuable</u>	<u>Method of improvement</u>
83.4	a/Better training for programers in superior programing practices.
80.8	Better training for programers in conversion problems and techniques.
77.8	Better training for installation

a/These were defined for the respondents in a previous question, which they were referred to.

- managers in conversion planning and estimating.
- 76.8 Ability to send the conversion team to appropriate specific training before starting work.
- 74.4 Required training for user/customer management to increase their awareness of the possible implications of their requests and deadlines.

---

Figure 13 shows that the supervisors value training highly in the area of conversion to foreign programs also. Again, the means can be regrouped:

- Better training in conversion (first and third choices).
- Better enforcement of documentation practices (second).
- More attention to conversion in selection of software (fourth).
- Better management (fifth).

Of these, the first and second are aimed at improving the original quality of the material to be converted, and the others are aimed at better conduct of the conversion effort itself.

---

Figure 13: Supervisors' major means of improving conversion of foreign programs:

<u>Percent ranking method valuable</u>	<u>Method of improvement</u>
86.6	Better training for programers in superior programing practices.
84.1	Better enforcement of good documentation practices.
83.3	Ability to send the conversion team to appropriate specific training before starting work.



- 81.1 More weight to conversion costs and problems in the selection of foreign programs to be converted.
- 80.4 Better training for installation managers in conversion planning and estimating.

Figure 14 shows that the programmers also value training highly. They selected four of the same five most important causes and ranked them somewhat differently. As with the supervisors' choices, the first is aimed at improving what will eventually be converted (better development), as is the third; the other three are aimed at better conduct of the conversion itself.

Figure 14: Programers' major means of improving conversion of foreign programs:

<u>Percent ranking method valuable</u>	<u>Method of improvement</u>
84.7	Better training for programers in superior programing practices.
80.7	Better training for programers in conversion problems and techniques.
79.1	Better enforcement of good documentation practices.
78.9	Better training for installation managers in conversion planning and estimating.
78.7	Ability to send the conversion team to appropriate specific training before starting work.



DEPARTMENT OF HEALTH, EDUCATION, AND WELFARE  
OFFICE OF THE SECRETARY  
WASHINGTON, D.C. 20201

JUN 10 1977

Mr. Gregory J. Ahart  
Director, Human Resources  
Division  
U.S. General Accounting Office  
Washington, D.C. 20548

Dear Mr. Ahart:

The Secretary has asked that I respond to your request for our comments on your draft report, "Federal Spending for Conversion of Computer Programs Could Be Reduced." We have carefully reviewed your report and have no substantive comments.

Thank you for the opportunity to review this draft report before its publication.

Sincerely yours,

  
Thomas D. Morris  
Inspector General



ASSISTANT SECRETARY OF DEFENSE  
WASHINGTON, D.C. 20301.

COMPTROLLER

MAY 13 1977

Mr. D. L. Scantlebury  
Director, Division of Financial  
& General Management Studies  
General Accounting Office  
Washington, D. C. 20548

Dear Mr. Scantlebury:

This is in reply to your letter to the Secretary of Defense regarding your report dated April 8, 1977, on "Federal Spending for Conversion of Computer Programs Could be Reduced," OSD Case #4597.

While we are somewhat less optimistic than you about the potential for hard dollar saving as estimated in your draft, we support the contention that sufficient benefit will accrue to warrant establishment of a conversion center, vigorous promotion of quality and standards in software development, and standardization of software tools.

Sincerely,

Joseph P. Welsch  
Deputy Assistant Secretary of Defense



**UNITED STATES DEPARTMENT OF COMMERCE**  
**The Assistant Secretary for Administration**  
 Washington, D.C. 20230

8 JUL 1977

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


Dear Mr. Eschwege:

We have reviewed the draft report to Congress, entitled "Federal Spending for Conversion of Computer Programs Could Be Reduced" and concur in the three recommendations contained therein.

With regard to the recommendation concerning establishment of a Federal software conversion center, we believe this subject was exhaustively studied by the Department of Defense and the General Services Administration. This center is currently being used by several Federal agencies, including this Department in our Employee Information System conversion project. [See note below.]

Concerning the recommendation that the National Bureau of Standards select a set of tools and techniques for Government-wide use to improve the productivity of computer programmers, NBS recognizes the need for greater use of productivity aids and will devote its efforts in this area as priorities dictate and resources allow.

Sincerely,

  
 Elsa A. Porter  
 Assistant Secretary  
 for Administration

GAO note: The Navy's Federal COBOL Compiler Testing Service is doing some conversion work, but as of this writing, there is no interagency agreement for-  
 mally establishing the conversion center itself.



UNITED STATES OF AMERICA  
GENERAL SERVICES ADMINISTRATION  
WASHINGTON, DC 20405



June 9, 1977

Honorable Elmer B. Staats  
Comptroller General of the United States  
General Accounting Office  
Washington, DC 20548

Dear Mr. Staats:

We have reviewed the GAO draft report entitled "Federal Spending for Conversion of Computer Programs could be Reduced", and offer the attached comments.

If there are any questions, please let us know.

Sincerely,

A handwritten signature in cursive script, appearing to read "Joel W. Solomon".

Joel W. Solomon  
Administrator

**Attachments**

**GSA COMMENTS  
GAO Draft Report**

**"FEDERAL SPENDING FOR CONVERSION OF COMPUTER PROGRAMS  
COULD BE REDUCED"**

- The GAO report identifies some of the major problems encountered in software conversion and recommends that a Federal Center be established for software conversion. In particular, it cites poor DP management practices, (conversion planning and training), enforcing the use of standard higher-level programming languages, and the lack of good documentation as contributing substantially to the software conversion problem.

Overall the report is a credible effort addressing a complex subject and we support its conclusions and recommendations. Yet its treatment of conversion is too narrow. Taken by itself, many of the observations and recommendations are true but the principal and underlying factor complicating the conversion issue is the lack of planning -- the management trade-off of major long term benefits at the expense of near term costs. Conversion and all its underlying symptoms (such as lack of documentation and unique programming features) are a result of a failure of management to adequately plan for future needs, even if it increases near term costs. The report should recognize this broader problem and note that conversion is but one of its sub-issues.

While adoption of the report recommendations would address some of the software conversion problems, it would not serve as a viable substitute for implementation of good DP management at the installation level.

- The report is supportive of efforts to establish the Federal Conversion Support Center. The proposed functions of the Center will more than satisfy GAO's recommendations. However, there are complex software conversion problems, not directly discussed in the report, which the Center will be confronting:

Conversion of systems which are comprised of various software programs developed by the incumbent equipment vendor, proprietary software firms, as well as the user agencies.

Conversion of efficient non-standard programs (such as proprietary data base management packages and to higher level standard languages which, if accomplished, could be extremely costly.

[See GAO note below.]

Although the Center will dramatically improve the present conversion environment, its establishment will not serve as a panacea and result in the immediate resolution of all Federal ADP conversion cost issues. Many of the problems confronted by the ADP community will require in-depth analyses and evaluations before the Government can expect to see the situation improved or the problems resolved.

- The report indicates that the use of unique programming features add considerably to the conversion effort required and that programmers should not use these features (or that at least they should be more closely controlled). While valid, the report fails to recognize that these unique features often provide programming and operational advantages over the near term.
- The report suggests that use of industry provided applications software could lower costs and speed implementation. This is the exception and not the rule. Generalized software packages can sometimes satisfy small applications requirements but frequently require modifications to fit specific needs.
- Data processing activities do not adhere to existing established Federal programming language standards. For example, although a Federal standard exists for COBOL, some data processing activities have waived this requirement and other(s) use vendor extensions to the standard COBOL, which make application software programs vendor dependent and incompatible with one another. These practices pose significant software conversion evaluation problems with any follow-on procurement. It would be helpful if this type of problem were addressed more fully in the report.

GAO note: Deleted material is not relevant to this final report.

[See GAO note 1.]

A copy of the questions included in the report would have been helpful. [See GAO note 2.]

[See GAO note 3.]

GAO notes:

1. Deleted comments are not relevant to this final report.
2. The questionnaires were omitted from the report to reduce its bulk.
3. Deleted comment is not relevant to this final report.





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

JUN 30 1977

Mr. D. L. Scantlebury  
Director, Division of Financial and  
General Management Studies  
U. S. General Accounting Office  
Washington, D.C. 20548

Dear Mr. Scantlebury:

We are in receipt of your draft report entitled "Federal Spending for Conversion of Computer Programs Could be Reduced" transmitted by your letter of April 8, 1977. Mr. Lance has asked that I provide you with OMB's comments.

We believe this report sheds some valuable light on a subject which has been of substantial concern to OMB over the last few years. We are particularly pleased that you have provided an estimate of both the total cost of software conversion (\$450 million) and an estimate of possible savings that can be achieved by proper management (24%). At a convenient point, I would like to have my staff gain a better understanding of the basis for this estimate so that we can work towards achieving the savings which you identified.

With respect to your recommendation that OMB support the establishment of a software conversion center, we agree. We understand that GSA and Navy are currently involved in negotiations to set up a center which would provide the services you recommend. One aspect of this recommendation, however, is unclear. We are uncertain whether you believe the center should, by regulation, be made the exclusive source for the various conversion services it will perform, i.e., estimates, contracting, performing conversion, or experience transfer. We have found that any publicly funded service monopoly has a tendency over time to degrade in performance and cost effectiveness unless it is subjected to some degree of competitive pressure. If this is intended in your recommendation we would suggest that this aspect be re-examined. We believe the Federal Computer Performance Evaluation and Simulation Center provides an excellent example of how a center of expertise can, through the force of competition, review itself and maintain efficiency. It

does not have exclusive jurisdiction over any of the services it provides but it has a steadily increasing demand for its services because of the quality of its products.

We would also call to your attention the views which OMB expressed in a letter to the Chairman of the Government Operations Committee (copy enclosed). OMB's responses to Committee recommendations 10 through 14 (See GAO note.) are particularly relevant to the subject matter of this report.

Sincerely,



Wayne G. Granquist  
Associate Director for  
Administrative Management

Enclosure

GAO note: The deleted recommendations are not relevant to the subject matter of this report.

January 19, 1977

Honorable Jack Brooks  
Chairman, Committee on Government  
Operations  
House of Representatives  
Washington, D.C. 20515

Dear Mr. Chairman:

This letter is in response to your letter of October 5, 1976 requesting OMB views on House Report No. 94-1746 entitled "Administration of Public Law 89-306, Procurement of ADP Resources by the Federal Government." Our comments and suggestions in regard to each of the 19 recommendations contained in the report are enclosed.

The hearings conducted by your Committee last summer and the subsequent report have served to identify a number of significant problems in the management of Federal ADP resources. These problems must be addressed and we have taken positive steps to do so.

Due to the significant impact which the recommendations could have on the future direction of Federal ADP management, OMB invited the larger ADP user agencies and a number of trade associations to provide comments on the Committee's recommendations. The responses which we received are enclosed for your information.

It is apparent from these responses that there is a strong consensus regarding the significance of the problems outlined in the report, but there is a wide divergence of views as to the causes of these problems. There are also strong differences of opinion regarding some of the solutions proposed. In short, there is agreement on the need to address these problems but no consensus on how to solve them.

In concluding the hearings on July 1, 1976, you indicated that "[o]nly time will tell whether these hearings will have a beneficial effect" and expressed the hope that "the report to follow will provide the necessary impetus

for the future administration of the law." We believe the report has achieved that initial objective. A number of recommendations have already been implemented or are in the process of being implemented. Where there are differences of opinion regarding some of the proposed solutions, we have either suggested alternatives or have taken steps to identify workable alternatives.

We would also like to stress that there is no quick solution to many of the complex issues which you have identified. For example, the development, coordination and implementation of standards, which are a prerequisite to the solution of a number of the problems identified, are not easily or quickly achieved in an area of rapidly changing technology. Similarly, the development, coordination and implementation of an effective long range planning process will not be brought about easily or quickly. These, and other basic reforms needed, will require a sustained effort by the Federal ADP Community accompanied by extensive consultation with the Congress, the private sector, State/local governments and others which would be affected by such reforms.

As OMB Deputy Director Paul O'Neill indicated during his testimony, we look forward to working in close cooperation with the Committee in furtherance of our mutual objectives of making effective use of computer technology.

With best wishes.

Sincerely yours,

/s/ James T. Lynn

James T. Lynn  
Director

Enclosures

cc: Secretary of Commerce  
Administrator, GSA

(See GAO note, p. 51)

**Recommendation 10 - NBS must develop necessary hardware and software standards to insure maximum economies and efficiencies in the procurement and utilization of ADP resources.**

OMB agrees with this recommendation. We believe that the Federal ADP standards program needs to be strengthened to achieve greater efficiency and economy. Such strengthening of the standards program must, however, be accomplished in a manner which will not stifle competition or preclude the Government from taking advantage of new technology.

Comments received from agencies and representatives of industry are very supportive of this recommendation and provide many constructive suggestions on approaches to achieving an appropriate balance between the sometimes conflicting objectives of reducing cost, exploiting technological development and encouraging competition.

A number of actions to strengthen the standards program have been initiated by OMB and NBS. In general, they focus on:

- ° A reappraisal of the goals and objectives of various ongoing standards development activities.
- ° A greater emphasis on defining the potential benefit of proposed standards or guidelines before undertaking the development of such standards or guidelines.
- ° A prioritizing of these activities so that resources can be focused on those activities which will produce the greatest benefits.

- A greater emphasis on clearly prescribing (in the applicable FIPS PUB) all policies, responsibilities and administrative actions required for appropriate implementation of each standard or guideline. These include:
  - Whether the standard is mandatory or voluntary.
  - Specific implementation actions which agencies are expected to take.
  - The conditions or criteria under which waivers to mandatory standards may be granted.
  - The procedures for granting waivers.
  - The time frame for implementation.
  - The requirements for use or disposition of existing inventories of nonstandard supplies, hardware or software.
  - Requirements for including the standard in procurements by GSA and/or the agency.
- Establishing a process for monitoring implementation of each standard and evaluating whether it is meeting its intended objective and, subsequently revising the standard, as appropriate, based on operational experience or changing conditions.
- A greater emphasis on evaluating technology trends and their implications in regard to the Federal ADP standards program.

Recommendation 11 - OMB must establish procedures for the effective enforcement of ADP standards and designate GSA as the agency to enforce compliance with such standards.

OMB is in complete agreement with what we believe to be the objective of this recommendation - to achieve compliance with mandatory standards - but only partially agrees with the proposed means for achieving this objective.

We believe that the Department of Commerce (NBS) under P.L. 89-306, E.O. 11717 dated May 9, 1973, and the policy guidance provided to the Secretary of Commerce by OMB in a letter dated December 15, 1966 has adequate authority for the Federal ADP standards program. In our view, NBS has the responsibility to develop and implement the type of standards program envisioned by P.L. 89-306 and E.O. 11717 and this would necessarily entail, among other things, the articulation of the goals and objectives of such a program (e.g., compatibility in equipment), the development of appropriate standards

or guidelines to achieve those goals and objectives, the development of the means for assuring the effective implementation of these standards or guidelines and the development of the means to monitor and evaluate the impact of such standards or guidelines. We believe that P.L. 89-306 contemplates that the exercise of this authority by the Department of Commerce (NBS) be undertaken in coordination with industry, the user agencies, GSA and OMB in order to ensure that the requirements of these entities are met and to permit effective implementation. Some questions have recently been raised by representatives of the Department of Commerce in regard to the scope of responsibility and authority in particular matters vested in the Department of Commerce (NBS) by P.L. 89-306, E.O. 11717 and related OMB policy guidance. We will explore and resolve these questions with the Department in conjunction with our plans to reissue OMB Circular No. A-71 as indicated in response to recommendation 3. We are satisfied that the standards or guidelines contemplated can be effectively implemented through existing authorities, but should the need for additional authority become evident, such authority shall be sought.

In our view, good management requires that overall responsibility for the Federal ADP standards program be retained by one organization and not be divided. It has been and continues to be our belief that the Department of Commerce (NBS) is the logical organization to carry out these responsibilities. Retaining the overall responsibility for the Federal ADP standards program within the Department of Commerce (NBS) does not, in our view, preclude the Department from coordinating, assigning, or delegating certain authorities for developing, implementing or monitoring compliance with individual standards to the agencies, GSA or others as appropriate. We believe the method for implementing individual standards will depend upon the nature of the standard. For example, we would agree that mandatory standards which affect products or computational services purchased by the Federal Government should be implemented largely through GSA procurement regulations. On the other hand, we question whether GSA should categorically be made responsible for "enforcing" all FIPS PUB's including those which provide guidance to agencies on the use or management of their ADP resources. Such assignment, depending on the nature of the standard, could be in conflict with the provisions of the Act which precludes GSA from interfering in agency use of ADP. Further study of this question is necessary.

We believe that determinations on the most appropriate means for implementing Federal ADP standards must be made on a case-by-case basis. We are working with NBS to establish a program for proceeding with analysis as outlined in our response to recommendation 10.

Recommendation 12 - All ADP programs should be converted to higher level languages except in those cases where OMB specifically determines that the national interest requires otherwise.

OMB agrees that more emphasis should be placed on agency use of high level languages since such languages facilitate the conversion of software from one hardware system to another and reduce the costs inherent in such conversions. However, more precise criteria must be established for determining when it is in the national interest to use high level languages, either in the conversion of existing software or in writing new software.

Agency comments indicated general agreement with the recommendation that greater use be made of high level languages but urged that mandatory use of high level languages be limited to new software. They expressed the concern that, even were it possible to do so, the conversion of all existing software to higher level languages would be very costly, unnecessary and counterproductive to the goals of economy and efficiency. Most agencies that commented on recommendation 12 felt that existing software should be converted only when it is economical and efficient to do so.

We believe that the appropriate way to implement this recommendation is to have NBS:

- ° Establish standards, or refine existing ones, on the use of high level languages for new software.
- ° Establish criteria to determine under what conditions existing software should be converted to high level languages.

We believe that this is a more appropriate way to approach the problem than having OMB review software on a case-by-case basis to determine if it is in the national interest to convert it. As indicated in our response to recommendations 10 and 11, the NBS guidance should include appropriate mechanisms for implementing and monitoring compliance with the standards. Given the advantages that can accrue from the use of high level languages we believe that NBS should place a high priority on efforts in this area.



Recommendation 13 - Software conversion costs should not be considered in the evaluation of bids for the procurement of ADP systems except to the extent that such costs involve direct out-of-pocket costs for program conversion.

OMB and the agencies have had some difficulty formulating a response to this recommendation since the meaning of the phrase "direct-out-of-pocket" costs is unclear. It is our understanding that this recommendation:

° Stems from a concern that agencies frequently justify less than fully competitive procurement on the grounds that software conversion would be too costly to consider alternatives requiring such conversion.

° Assumes that the implementation of recommendation 12 will significantly reduce software conversion costs.

° Is intended as an interim measure to enhance competition until such time that recommendation 12 is implemented.

As indicated in our response to recommendation 12 we endorse the use of high level languages and the development of appropriate standards for converting existing software and writing new software. The approach recommended by OMB would permit the orderly conversion to high level languages in an economic and efficient manner over a period of time. This approach and the other steps we are taking, such as promoting functional specifications, promoting research on standardized data base management systems and improving long range planning, should reduce the costs of conversion and lead to fuller competition in ADP procurements.

OMB believes that this recommendation should not be implemented prior to the conversion of most software to high level languages. If agencies were required to ignore the costs of software conversion in the current environment they might be placed in the position of having to select a significantly more costly alternative without any offsetting public benefit. This would, of course, be contrary to the P.L. 89-306 goal of economy. OMB believes that ADP procurements should be conducted in a manner such that each ADP system is evaluated on the basis of its total system life cycle costs, including software conversion costs and the least cost alternative is selected. Generally, the agencies took issue with the Committee's recommendation and expressed the view that all costs associated with converting from one system to another should be considered and that it would be unfair to have an agency absorb software conversion costs.

We believe that it is GSA's responsibility to develop a costing methodology for ADP procurements which will provide guidance on what cost elements should be considered in comparing alternative systems and under what conditions specific costs may be excluded. We would urge GSA to work closely with the Committee staff in order to assure that the staff's intentions regarding direct out-of-pocket costs are fully considered. We also understand that the GAO has undertaken research in this area. When the results of this research become available GSA should review the GAC efforts and take action to implement appropriate recommendations.

Recommendation 14 - Research should be undertaken pursuant to OMB's direction aimed at the development and use of management information systems which contribute to competitive ADP procurements and utilization.

OMB agrees that research relative to the development and use of standardized data base management systems should be undertaken. Standardized data base management systems should ultimately facilitate the conversion of data files from one hardware system to another and lead to greater opportunities for competitive procurements.

P.L. 89-306, E.O. 11717 and the OMB policy letter of December 15, 1966 assign the responsibility for the Federal ADP standards program including ADP research to the Department of Commerce. The National Bureau of Standards has recently created a task force to study problems associated with data base management systems. We will work closely with NBS in this area.

(See GAO note, p. 51)

(91306)