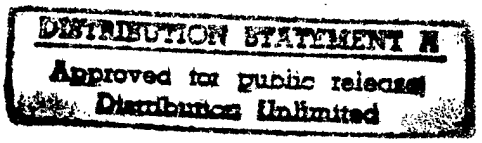
U.S. DEPARTMENT OF COMMERCE
National Technical Information Service
PB80-105638

Contracting for Computer Software Development Serious Problems Require Management Attention to Avoid Wasting Additional Millions

(U.S.) General Accounting Office, Washington, DC



9 Nov 79

BY THE COMPTROLLER GENERAL

Report To The Congress

OF THE UNITED STATES

Contracting For Computer Software Development--Serious Problems Require Management Attention To Avoid Wasting Additional Millions

Federal agencies often hire contractors to develop computer programs and associated documents. Many development contracts are successful, but some result in unsatisfactory or useless software.

Several causes of problems were common to all contracts GAO reviewed that encountered difficulties. This report recommends that the National Bureau of Standards and the General Services Administration issue specific guidelines to assist Federal agencies in recognizing and dealing with the unique problems of contracting for software development.

FGMSD-80-4
NOVEMBER 9, 1979

BIBLIOGRAPHIC INFORMATION

PB80-105638

Contracting for Computer Software Development--Serious Problems Require Management Attention to Avoid Wasting Additional Millions.

9 Nov 79

PERFORMER: General Accounting Office, Washington, DC. Financial and General Management Studies Div. FGMSD-80-4

Federal agencies often hire contractors to develop computer programs and associated documents. Many development contracts are successful, but some result in unsatisfactory or useless software. Several causes of problems were common to all contracts GAO reviewed that encountered difficulties. This report recommends that the National Bureau of Standards and the General Services Administration issue specific guidelines to assist Federal agencies in recognizing and dealing with the unique problems of contracting for software development.

KEYWORDS: *Government procurement, *Software engineering.

Available from the National Technical Information Service, Springfield, Va. 22161

PRICE CODE: PC A05/MF A01

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

Contracting for computer software development can be an effective alternative to software development by Federal employees. However, the effectiveness of such contracting depends on very careful contract management.

This report discusses the problems that Federal agencies have encountered in contracting for computer software development and recommends means of improving such contracting.

We are sending copies of this report to the Secretary of Commerce and to the Administrator of General Services.

Comptroller General
of the United States

COMPTROLLER GENERAL'S
REPORT TO THE CONGRESS

CONTRACTING FOR COMPUTER SOFTWARE DEVELOPMENT--MORE MANAGEMENT ATTENTION COULD AVOID WASTING ADDITIONAL MILLIONS

D I G E S T

Many Federal agencies have computer programs--called software in the data processing industry--developed by outside sources. These sources may be either private firms or other Federal agencies.

Although new software is often developed successfully by outside sources, GAO found that too many contracts experience large cost overruns and lengthy delays, and agencies may be dissatisfied with the product.

Agencies contract with outside sources for custom software development for various reasons. For instance, they do not have enough staff or expertise to develop it themselves, or they can get it at lower cost.

GAO sent questionnaires to 163 software contracting firms and 113 Federal project officers who had experience with software development contracts to identify what had caused trouble and what might be done to improve development efforts. Certain things causing problems for both contractors and agencies were common to all reviewed contracts that had trouble.

GAO examined nine cases of software development in detail. Eight had problems, but their overall performance cannot be taken as representative--some came to GAO's attention because they were failures. Nevertheless, the cases illustrated many of the same causes of difficulty that GAO's questionnaire respondents had identified.

Only one of the nine cases yielded software that could be used as delivered. The combined total costs and development times of

the nine cases increased from estimates of $3.7 million and 10.8 years to an actual cost of $6.7 million and an actual duration of 20.5 years.

COMMON CAUSES OF SOFTWARE DEVELOPMENT CONTRACTING PROBLEMS

Federal agencies contract for software development with little specific guidance. This circumstance was common to almost all reviewed contracts. (See p. 15.)

Agencies also overestimate the stage of system development they have reached before they contract. They overestimate the completeness of their own work, such as analyzing user requirements, before they contract for software development. Often, an agency's preliminary work is inadequate and must be done again by the contractor.

Overestimating its own preliminary work can lure an agency into issuing inappropriate contracts and using inadequate criteria for contractor performance. (See p. 17.) By failing to stipulate what constitutes satisfactory performance by the contractor, agencies reduce the likelihood that the delivered software will be satisfactory. The lack of a good contractual description of what the contractor is to do makes it difficult for the agency to claim poor contractor performance. (See p. 20.)

Agencies quickly overcommit themselves and fail to control contractors through strict phasing. They will sometimes commit themselves to the entire software development, including writing, testing, and delivering the computer programs before they even have the user requirements--what the software is to do--clearly identified. In such situations, a phased contract, initially committing the agency only to an analysis and design phase, and then proceeding only if the first phase proves satisfactory, would be much more suitable. (See p. 21.)

Agencies do not manage software development contracts during execution. Management failures while the work is being done included

excessive changes and afterthoughts, failure to inspect intermediate stages of the work, and failure to require progress reports from the contractor. In one case, officials stated that they could not review the work because it was being done on the contractor's premises. (See p. 23.)

In contracts that have problems, contractual testing requirements are often sketchy or absent. Agencies accept and pay for software without adequately inspecting and testing it. Contractors identified inadequate agency test data as a frequent source of problems. Failures to inspect test output and documentation products also occur. In one case, the contract called for the use of one programming language but the delivered programs were written in another. The contractor still got paid. (See p. 24.)

Some problems occur because agencies fail to establish a single focal point for communication with contractors. Communications difficulties and delays occur when contractors have no identified single source for answers or proposed changes and interpretations of requirements. (See p. 25.)

GAO also found that problems arise because agencies do not adequately specify or enforce contract clauses for recovery in the event of poor performance by the contractor, and contractors frequently fail to provide adequate software documentation.

RECOMMENDATIONS

GAO recommends that the Secretary of Commerce, through the National Bureau of Standards, and the Administrator of General Services issue specific guidelines to assist Federal agencies in recognizing and dealing with the unique factors added to custom software development when it is done by contract. The following areas should be covered:

--Internal agency management practices necessary to write, manage, and monitor software development contracts.

--Specific instructions on how to tailor software development contracts to the state of system development that an agency is in at the time it lets a contract.

--Guidance on contract stipulations regarding the phasing of the software development.

--Guidance on the review and approval procedures agencies should follow at the end of each phase of software development.

--Guidance on performance specifications to be included in the contract to clarify quality requirements for the software.

--The importance of requiring the software contractor to have a formal quality assurance program that is documented and subject to audit.

--The degree of definition required to properly define such things as (1) documentation standards, (2) adherence to programming language standards, (3) acceptance testing procedures, and (4) satisfactory performance by the contractor.

--How to handle changes in the software being developed with minimal disruption.

--How to ensure that the contractor follows sound system development practices.

--The effective use of contract clauses which would deny payment in case of poor performance by the contractor.

The above recommendation could be achieved to a large extent if the National Bureau of Standards and General Services Administration designed a series of model contracts containing detailed clauses on such items as documentation, phasing, and testing. A full explanation of their need and value should accompany these clauses. Agencies could extract relevant clauses and construct contracts to fit their particular situations. Such model contracts are recommended primarily as aids to agency software development

contract management, but they would also aid the agency procurement function in general.

GAO also recommends that Federal agencies involved in software development contracting train project managers in such overall skills necessary to manage those contracts as software, contracting, and management. Agencies should also take appropriate action in each phase of software development contracting. GAO offers a provisional checklist for contracting for software development to outline appropriate action to be taken for each phase (See app. I.)

In written comments on the report, the General Services Administration and the Department of Commerce generally agreed with the conclusions and recommendations. (See p. 31.)

Contents

		Page
DIGEST		i
CHAPTER		
1	INTRODUCTION	1
	Federal use of computers and software is extensive	1
	Federal agencies spend a lot on contracting for software development	2
	Roles of various agencies	3
	Scope of review	3
2	SOME SOFTWARE DEVELOPMENT CONTRACTS EXPERIENCE EXTRA COSTS, DELAYS, AND OTHER PROBLEMS	5
	Software development traditionally has been a problem area	5
	Contracting for software development adds to its difficulty	7
	Too many software development contracts have problems	8
	Software development contracting problems have undesirable effects	12
3	MAJOR CAUSES OF PROBLEMS IN CONTRACT SOFTWARE DEVELOPMENT	15
	Agencies now contract for software development with little guidance	15
	Software development contracts in trouble have common causes	17
	Agencies and contractors agree on some causes of problems and ways to improve contracting	26
4	CONCLUSIONS, RECOMMENDATIONS, AND AGENCY COMMENTS	28
	Conclusions	28
	Recommendations	29
	Agency comments	31
APPENDIX		
I	Provisional checklist for software development contracting	33

II	Case studies in contracting for computer software development	46
	The payroll system that never paid anyone	47
	The $2-million, contractor-developed management information system that was replaced by a $20,000 in-house system	51
	The payroll/personnel system that wasted $1 million	56
	The failure to develop a centralized accounting system	59
	The successful compiler development	63
	The test scoring program that failed its own test and had to be rewritten by agency programmers	66
	The three-party contract that failed to yield a useable accounting system	70
	The engineering technical information system that was developed from incomplete technical information	74
	The court case retrieval system	76
III	Agency comments	79

ABBREVIATIONS

ADP	automatic data processing
COBOL	common business oriented language
GAO	General Accounting Office
GSA	General Services Adminstration
NBS	National Bureau of Standards
OMB	Office of Management and Budget

CHAPTER 1

INTRODUCTION

FEDERAL USE OF COMPUTERS AND SOFTWARE IS EXTENSIVE

The Federal Government is the world's largest user of automatic data processing (ADP) resources, incurring costs that have been estimated at over $10 billion per year and which continue to increase. The General Services Administration's (GSA's) ADP summary for fiscal 1978 reported that the Government owns or leases over 12,100 computers. These computers are used to process a variety of applications ranging from delivering health and welfare services, to administering social security and veterans' benefits, to exploring space, to analyzing and reporting on the military.

Computer programs--generally referred to in the industry as software--are what make all these computers run. A computer without programs is like a phonograph without records--it won't play.

Recent industrywide estimates predicted that organizations, rather than have their employees write computer programs, would increase their spending for readymade programs. For example, indications are that in 1978, public and private organizations may have spent about $2.6 billion on programs developed by outside sources. If realized, this amount will be a 27-percent increase over 1977. We estimate that about 800 independent software suppliers operate in the United States, and experts have predicted a five-fold increase in jobs in the software industry by 1985. Some of the software suppliers sell ready-made software; others contract to develop custom-built software for clients. 1/ We have numerous indications that having software developed by outside sources and using ready-made software are accepted and successful practices in the private sector.

Computer software development has historically been a problem and is further complicated by contracting for it. Literature on the subject contains many discussions of software development projects that were late, cost too much, or

---

1/The difference between readymade and custom-developed software is analogous to the difference between a ready-to-wear suit and a made-to-order suit.

failed completely. Causes cited for these problems have included the difficulty of defining the work to be done, changes to the scope of the work during the project, and lack of communication between ADP users and computer specialists.

FEDERAL AGENCIES SPEND A LOT ON CONTRACTING FOR SOFTWARE DEVELOPMENT

Although few reliable statistics are available, it is estimated that Federal agencies contract for several hundred million dollars of computer software development by software vendors and by other Government agencies annually. For example, the GSA ADP summary reported that about $254 million was spent in fiscal 1978 for contract systems analysis, design, and programming in the general management category alone. 1/

Federal agencies usually contract for software development because they lack the staff or skill to develop it in-house. The goal of such contracting is to promptly deliver computer programs which (1) automate necessary tasks for the agency, (2) are usable as delivered, (3) have reasonable operating and maintenance costs, and (4) are written so they can be easily modified later to meet changing requirements. Reasonable operating and maintenance costs require that the programs be skillfully written (to minimize their costs), thoroughly tested for correctness, and well documented for ease of operation and interpretation. If later modification is to be made with ease, the programs must be well documented for the maintenance programmers who will modify them.

We undertook this study to determine the extent of problems in Federal software contracting, to identify some of the common underlying causes of problems in this area and, if possible, to recommend means of improvement.

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 is responsible for procuring and maintaining Federal ADP resources. GSA receives technical advice from the Secretary of Commerce, primarily through the

---

1/This figure includes contract services for which there was no deliverable product as well as for contracts with a product, but does not include software developed for special-purpose computers, such as those included in weapons systems.

National Bureau of Standards (NBS), and both of 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 development as a frequent and expensive activity. 1/ Our past reports to the Congress have recommended improvements in ADP management both governmentwide and at specific agencies.

SCOPE OF REVIEW

In this review, we concentrated on contracts for the development of custom-built computer programs for business and administrative purposes. The exception is our fifth case in which the contractor developed a compiler. 2/ This review did not address military or special management contracts, such as those in which the computers and their programs are parts of a weapons system, nor did it address readymade software purchases.

We did the following work in our review.

--We administered two nationwide questionnaires: one to 163 software contractors, the other to 113 Federal data processing personnel with software contracting experience. The questionnaires asked contractors and agency personnel to (1) identify those problem areas they experienced in software contracting, (2) identify the causes of those problems, and (3) evaluate proposed solutions.

--We identified and analyzed nine cases where software development was contracted for with Federal funds. Some were brought to our attention because they were problem cases. We evaluated them and attempted to determine the causes of any problems identified. Narratives on the cases are included in appendix II.

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

2/A compiler is a computer program which translates statements in a programming language (written by a person) into the internal code of the computer. The results of such translation is what actually makes the computer perform the required tasks.

--We contacted NBS and GSA to determine what guidance they had issued to Federal agencies to help them successfully contract for software.

--We examined agencies' internal guidelines to determine the adequacy of procedures to be followed in contracting for computer software development.

--We researched recent studies and publications.

--We developed a provisional checklist of agency management actions to improve software development contracting. We feel that this checklist will be helpful to those preparing to contract for software development. (See app. I).

CHAPTER 2

SOME SOFTWARE DEVELOPMENT CONTRACTS EXPERIENCE

EXTRA COSTS, DELAYS, AND OTHER PROBLEMS

We found that some software development contracts made by Federal agencies have experienced dramatic cost and time overruns, user dissatisfaction, and, sometimes, the contracts resulted in software that never worked in spite of extra time and money spent on it by agency programmers after the contractor had left.

In this chapter, we will discuss the characteristics of custom software development, why contracting for it adds to its problems, and the conditions under which contracting is done that we found in our investigation.

SOFTWARE DEVELOPMENT TRADITIONALLY HAS BEEN A PROBLEM AREA

Software development

The complete life cycle of computer software is divided into requirements analysis, system design or specification, development, and operation. Development time ends when the software is in production. 1/ Traditionally, many software development projects have experienced one or more of the following problems:

--They have cost more than expected and have run late. Adding more people to late projects has not often restored them to schedule.

--The first production versions delivered were really prototypes in conventional engineering terms. Besides the programs themselves being prototypes, their documentation was often sketchy or missing.

--Because prototypes were delivered, the operational costs of fixing and modifying them have typically been as great as the development costs and sometimes far greater.

1/Software that is in production is doing the task for which it was written; for example, a payroll program is in production when it is printing checks.

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 new programmers, most of whom were self-taught on the job; and frequently, low productivity and poor product quality resulted. Third, there was little idea then of how to train programmers properly. Fourth, a tradition grew that programmers were secretive craftspersons whose products, during development, were their own property.

Software operation

The operational phase lasts from the beginning of production to the time the software is replaced or discarded. During this time, costs beyond normal operating costs may be incurred to (1) correct errors in the software, (2) modify the software so that new functions can be added, (3) tune the software to reduce excessive operating costs, or (4) convert the software to run on another computer.

Corrections of errors and modifications are commonly combined under the term "maintenance." Estimates of programmer time spent maintaining software that is developed traditionally have ranged in numerous organizations from 20 to 80 percent of total programming effort.

Historically, projects both to develop original software and to convert current software have often been completed later or at higher cost than predicted, or both. Many causes contribute to this situation--some are managerial, some sociological, and some technical.

In the managerial category, the ability to measure and predict software projects has been lacking. Workers who produce software are commonly considered to be difficult to retain and to manage effectively. Customers for whom the software is created (end users) often do not have a clear understanding of the process or function they want automated when the software development begins. Changes requested after projects have started, which seem trivial to the customers, have often required major rework and have resulted in delays and increased costs.

In the sociological category, our previous software conversion review 1/ found that many computer programmers

---

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

view themselves as craftspersons, with strong feelings that "I'd rather do it myself." Traditionally, the programmers' secretive attitude was indulged by management because programmers were scarce and difficult to retain.

Technical problems result from the need to meet deadlines--programs are often designed and written hastily, and are tested and documented inadequately or not at all. Thus, quality is sacrificed to urgency. Documentation--material prepared to explain a computer program--is often deferred until after the program is running and sometimes is never completed. When programs are later modified or converted, the work is usually done by someone other than the originator. If documentation is missing, incomplete, or obsolete, a great deal of the original development work often must be repeated by the person modifying or converting the program before that person can hope to modify it or convert it successfully.

CONTRACTING FOR SOFTWARE DEVELOPMENT ADDS TO ITS DIFFICULTY

The development of original software which meets users' needs has a tradition of managerial and technical difficulty even when the programmers and analysts developing it work for the same organization as the users who need it. Several sources of difficulty, as described below, are added when the software is developed by outsiders, as it is when Federal agencies contract for software development.

--The problem definition and/or user requirement must be defined so that "outsiders" can understand it.

--Contracting introduces an extra communication link between the software users and developers.

--The capability of the contractor should be checked and verified before the contract is let.

--Extensive acceptance testing criteria and test data should be developed before the contract is released and acceptance tests contractually required. Testing is also needed with software developed in-house, but contracting emphasizes the need for it.

--Contractor personnel must be informed about agency operations.

--Agency management must control the quality of work done outside the agency.

--First-hand observation of progress is more difficult for the agency.

--The contractor's knowledge of the software is lost when the finished software is brought in-house. Even with good documentation, which is sometimes lacking, agency personnel must become familiar with the programs.

--Acquisition of software from a contractor requires an agency to identify and meet all applicable Government procurement regulations.

TOO MANY SOFTWARE DEVELOPMENT CONTRACTS HAVE PROBLEMS

Our sources of information indicated that many software development contracts produce software that is useful to the customer. However, a substantial number do not. Also, even some of the delivered software that can be used must be reworked by agency staff after delivery before it works satisfactorily. Those who responded to our questionnaire indicated that problems often occurred with development contracts. Our case studies confirmed these problems and illustrated the situations in which they occur.

Questionnaires reported common problems

Figures 1 through 4 show the frequency 1/ that the questionnaire respondents reported for several conditions. Figure 1 shows what the respondents said about dollar overruns: 21 percent said that their occurrence was "very common," 29 percent said "fairly common," and only 6 percent would say they "never" occurred. Figure 2 shows that our respondents reported frequent calendar overruns: 30 percent said they were "very common," 32 percent said "fairly common," and only about 2 percent would say they "never" happened. Figure 3 shows that our respondents reported that even though software is finally delivered, it must often be reworked: about 9 percent said that problem was "very common," 35 percent said "fairly common," and only 6 percent said it "never" occurred. Figure 4 indicates the responses to the worst situation--that is, software was paid for and not used, for which only 20 percent said it "never" occurred.

1/In this question, we defined very common as over 75 percent of the time, fairly common as 51 to 75 percent of the time, and not very common as 25 to 50 percent of the time.

Figure 1

"Software development has dollar overrun."

Response	Respondents	
	Number	Percentage
Very common	24	21.2
Fairly common	33	29.2
Not very common	29	25.7
Very rare	11	9.7
Never occurs	7	6.2
Don't know	9	8.0
Total	113	100.0

Figure 2

"Software development has calendar overrun."

Response	Respondents	
	Number	Percentage
Very common	34	30.1
Fairly common	36	31.9
Not very common	29	25.7
Very rare	9	8.0
Never occurs	2	1.8
Don't know	3	2.7
Total	113	a/ 100.0

a/Does not add due to rounding

Figure 3

"The delivered software must be corrected or modified by in-house programmers before it is usable."

Response	Respondents	
	Number	Percentage
Very common	10	8.8
Fairly common	39	34.5
Not very common	40	35.4
Very rare	15	13.3
Never occurs	7	6.2
Don't know	2	1.8
Total	113	100.0

Figure 4

"The software is paid for but never used."

Response	Respondents	
	Number	Percentage
Very common	-	-
Fairly common	4	3.6
Not very common	18	16.1
Very rare	64	57.1
Never occurs	23	20.5
Don't know	3	2.7
Total	112	100.0

Case studies confirmed questionnaires

We examined nine cases in detail, which included visits to both agency and contractor sites, examining documents, and interviewing those persons involved who could still be reached. We must caution the reader that several cases came to our attention because they were problems. Therefore, although they cannot be taken as representative of all Federal software development contracts, they nevertheless dramatically illustrate software development contracting problems.

Our cases also provide examples of the common problem conditions reported by the questionnaire respondents. Figure 5 summarizes the nine cases in terms of what the Government got for its money--less than 2 percent of the total value of the contracts could be used efficiently as delivered. Also, these contracts averaged delays ("calendar overruns") of about 75 percent of their original time estimates. Figure 6 summarizes conditions that were common to the case studies and shows how they relate to the conditions identified on the questionnaires.

Figure 5

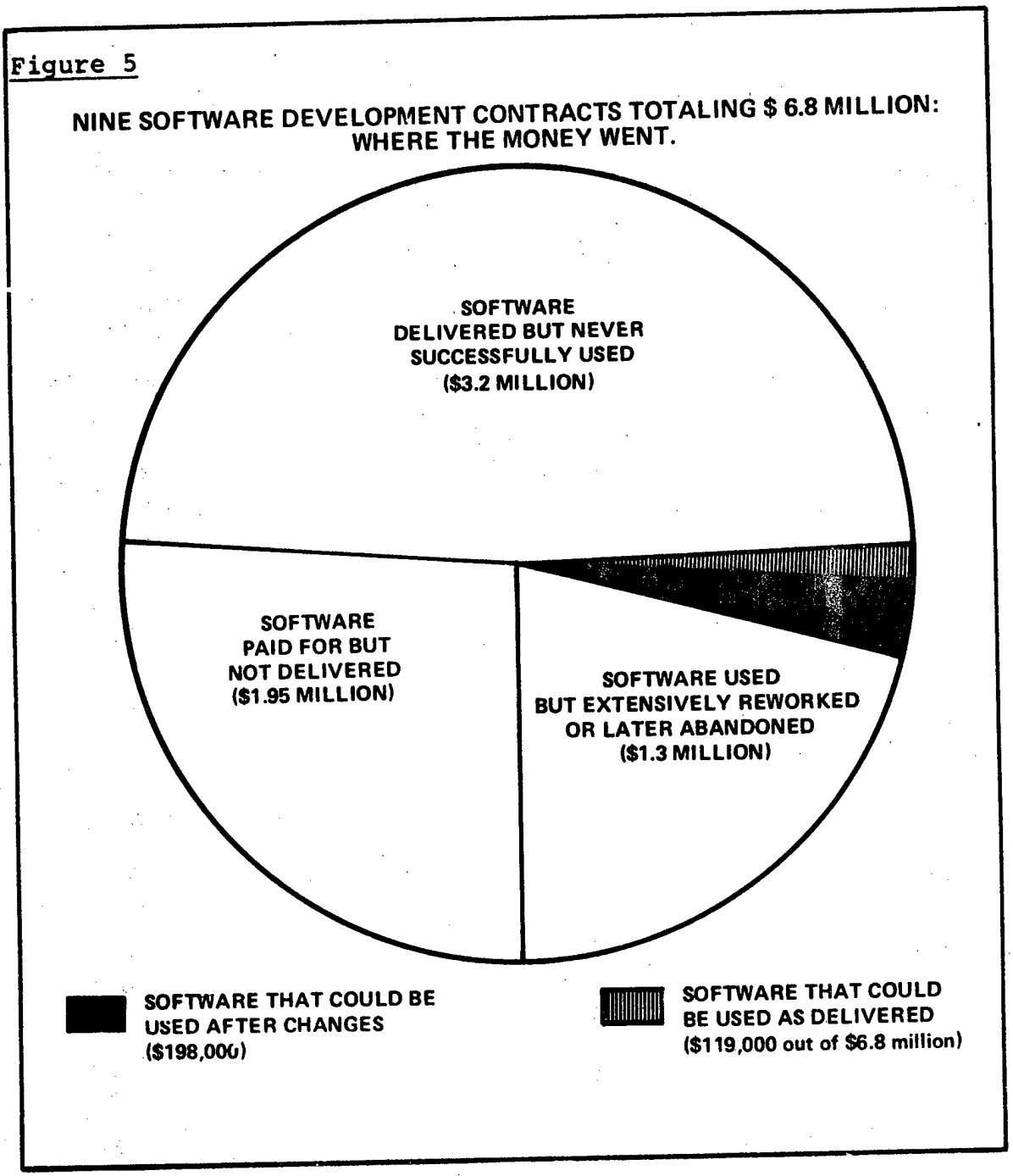


Figure 6

Software development contracting cases which had problems identified on the agency questionnaire

Condition identified	Case Number 1	2	3	4	5	6	7	8	9
Software development has dollar overrun	x	x	x	x				x	
Software development has calendar overrun	x	x	x	x		x	x		x
The delivered software must be corrected or modified by in-house programmers before it is usable							x		x
The software is paid for but never used	x	x	x	x		x			
Attempts to re-work and/or use the software in-house were tried but failed	x					x	x		
Software met acceptance testing as delivered					x				

Our questionnaires and our case studies together show that problems in software development contracting occur frequently. We feel that too many such contracts have serious problems and that the situation could be improved significantly.

SOFTWARE DEVELOPMENT CONTRACTING PROBLEMS HAVE UNDESIRABLE EFFECTS

Serious problems in software development contracting have both immediate and delayed bad effects. The immediate effects can include:

--Delay of user functions. A user function, such as printing employee paychecks, is the reason why a software development is undertaken. If the software development is late, the user function may not be

automated at all, or may continue to be done with older, more costly software that was supposed to be replaced.

--Additional cost before delivery. The contractor often gets paid more than predicted.

--Additional cost right after delivery. Even when a contractor delivers software, in-house programmers may need to modify it before it is satisfactory for production. This modification adds cost and may fail.

The delayed effects which can occur include:

--Years of maintenance and modification that cost more than they should when the delivered programs are poorly organized or poorly documented.

--Slowed response to later requests for modification. Later requests for modification to what the software does for the users can come from the users themselves or can be legislated. Those requests are delayed when programs are poorly organized or poorly documented.

Figures 7 and 8 show what the Federal questionnaire respondents said about two indicators of delayed effects. Figure 7 shows that many of our respondents report difficult-to-modify software as a frequent occurrence. Figure 8, while giving a somewhat more optimistic picture, still shows a significant occurrence of difficult-to-understand software.

Figure 7

"The delivered software is difficult to modify."

Response	Respondents	
	Number	Percentage
Very common	6	5.3
Fairly common	42	37.2
Not very common	43	38.1
Very rare	13	11.5
Never occurs	5	4.4
Don't know	4	3.5
Total	113	100.0

Figure 8

"The contractor's programming practices are such that the software is easily understood by agency programmers."

Response	Respondents	
	Number	Percentage
Very common	16	14.2
Fairly common	71	62.8
Not very common	17	15.0
Very rare	7	6.2
Never occurs	-	-
Don't know	2	1.8
Total	113	100.0

CHAPTER 3

MAJOR CAUSES OF PROBLEMS IN CONTRACT SOFTWARE DEVELOPMENT

Among the software development contracts that have problems, we found several common causes of those problems, even though each case has unique features of its own. The responses to our questionnaire and our review of the case studies brought several of these causes to the surface.

We found that agencies contract for most software development because (1) they lack enough staff, or staff with the right skills, to do it in-house or (2) because they need the software sooner than in-house staff could develop it. We have numerous indications from this and other studies that agencies would generally develop software in-house if the resources were available. However, a cost advantage is sometimes used to justify contracting.

AGENCIES NOW CONTRACT FOR SOFTWARE DEVELOPMENT WITH LITTLE GUIDANCE

We found that agency staff connected with software development contracts typically have little guidance, either from central agencies or from their own agency headquarters.

Central agency guidance

The basic responsibilities of the central agencies are described in Public Law 89-306, the Brooks Act. The Office of Management and Budget provides fiscal and general oversight of ADP activity. OMB has delegated responsibility to GSA for attaining cost effectiveness in the selection, acquisition, and utilization of ADP resources. The National Bureau of Standards is assigned the task of developing technical standards and guidelines. OMB guidance--since the Brooks Act was passed--has indicated that NBS is also responsible for investigating the conduct of system studies, including (1) monitoring their performance and implementation, (2) preparing proposals, specifications, and system requirements, and (3) continually evaluating installation and system performance.

We asked NBS and GSA what guidance they had provided Federal agencies in the specific management aspects of software contracting. NBS representatives informed us that while their responsibilities involve management and contracting activity, their agency's emphasis has been and will continue to be on the technical aspects, such as the standardization

of the common business-oriented language (COBOL) for Government use. They provided us with a primer for project management and quality control, which, although it makes general references to contracting situations, for the most part contains system development instructions.

GSA's guidelines pertaining to the management of ADP resources have been issued in the form of Federal Property Management Regulations, subpart 101-32, while policies and procedures on procurement of and contracting for commercially available software are set out in Federal Procurement Regulation 1-4.11. Our review of these guidelines showed they deal almost entirely with procurement of commercially available software and not the specific management of contracting for custom software development. Like NBS, GSA has also issued system development or project management publications for agency use, but they do not deal specifically with software development contracting.

Guidance at the agency level

We asked the agencies involved in our case studies about the guidance--if any--they have issued on the management aspects of contracting for software development. Some agencies furnished us policy manuals and directives on software development. The manuals and directives were primarily instructions on how to develop systems, with little specific information on how to contract for system development.

Only one agency reviewed differed from this general pattern. This agency has issued a series of software acquisition management handbooks which are used internally as guidelines when contracting for software. They cover such functions as (1) monitoring and reporting of software development status, (2) statement of work preparation, and (3) software quality, cost estimation, and measurement. These guidebooks are a step in the right direction toward more specific management guidance on software contracting. We feel that they could serve as a basis for publications for NBS to disseminate governmentwide.

SOFTWARE DEVELOPMENT CONTRACTS IN TROUBLE HAVE COMMON CAUSES

Besides the lack of specific guidance discussed above, our case studies and the responses to our questionnaire identified several factors which were at the root of the problems with many software development contracts. The presence of those factors in our case studies is summarized in figure 9, and the cases are detailed in appendix II. Often, several factors were present in one case.

Agencies overestimate the stage of system development they have reached before the contract

By "stage of system development," we mean the point the agency has reached in the series of steps followed in developing a software system. For a number of reasons, it is critical that the agency be accurate in determining the steps it has completed before it begins the contracting process.

--If the agency completes some system development work before contracting, the contractor will presumably begin from that point. This exact point must be determined to get realistic cost and time estimates. If, after the contract is underway, the agency's previous work is discovered to be less advanced than initially thought, and the scope of the contractor's work increases, costs may increase enough to destroy any cost/benefit that may have originally justified the contract.

--The point to which system development has progressed may affect the type of contract that should be let. For instance, if the agency has completed the detail design of its system, a firm-fixed price contract for the programming work may be possible. On the other hand, if the agency has not defined its requirements and has no system design in mind, it might need to let a phased, cost-plus-fixed-fee contract with proper audit clauses because the exact scope of work is not known.

--If work done by the agency before the contract is let is found to be inadequate after the contractor begins work, the contractor often needs to begin again with a different approach. When this happens, the tendency is to try to save as much as possible of the work already done and to modify it to fit the new approach. This approach nearly always compromises the new system and makes it less efficient, causing higher operating costs.

--If the original work scope must be revised, it may call for skills the contractor does not have. For instance, a contractor may agree to do programming, only to find that design work is needed before programming can be started. The contractor may not be qualified to do the necessary design work.

Figure 9

Causes of problems that were present in our cases

Cause	Case number 1	2	3	4	5	6	7	8	9
Agency overestimated its own state of progress when it let the contract	x	x				x		x	
Incorrect agency management action, such as using inappropriate contract			x	x		x		x	
Agency failed to specify requirements adequately	x	x		x		x	x	x	x
Agency overcommitted itself							x		x
Agency failed to manage during execution, including excessive changes	x	x				x	x		x
Agency failed to adequately inspect and test	x	x	x			x		x	

In several of the case studies, the agencies accomplished considerably less than they thought they had when they let the contracts. The most obvious effects were delays and extra costs because the contractor had to perform unexpected extra work. Other effects were noted which substantiated the importance of an agency isolating precisely how far it had progressed in its system development before contracting. For instance:

--In Case 1, the system concept the agency originally proposed was changed long after the contract commenced. An analysis of the product delivered under this contract showed that so much of the old concept remained in the new system that it was complicated and inefficient.

--In Case 2, a contract was let to a contractor because he was experienced in developing a similar system for the agency, and some of the programs in the earlier system were to be adapted to the new one. However, the agency did not establish the technical feasibility of adapting the old programs before letting the contract, and when they found that the old system could not be adapted and a new design concept was needed, the experience for which the contractor was hired was useless to the new approach. As a result, the agency's system development status reverted to an earlier stage, and the scope of the contract became far greater than originally intended.

These and other cases provide examples of agencies' failure to determine their correct system development status. (See app. II.)

Agencies take incorrect management action

Incorrect actions by agency management can condemn a software development contract before it is even started. As discussed above, sometimes the incorrect actions are based on the agency's overoptimistic assessment of the work it has done.

This was the case in our third, fourth, sixth, and eighth studies. In the third case study, the agency awarded a fixed-price contract with phased development but did not require that each phase be approved before work was begun on the next. This type of contract was awarded even before the the user requirements had been identified--a situation for which a fixed-price contract is not appropriate. In the fourth case, the agency let the contractor define the criteria by which his own work would be judged. Agency management also failed to derive an agency consensus on system requirements. This failure allowed an excessive number of reports to be required--188--as compared to much smaller numbers of reports required in several other accounting systems for which 44 was the largest number.

In our sixth case, the agency management relied upon discussions with the contractor to define requirements and stated in the contract that written progress reports would not be necessary. Since no agreement about the requirements was put in writing, the agency later had no way to require the contractor to perform. In our eighth case, the agency management failed to include standards, and criteria were written without adequate preliminary systems analysis work.

Failure to specify what constitutes satisfactory performance

It is important to specify not only what the software is to do but how well it is to do it and how well it is to be described. Whether the specifics of the software are known or not, some general requirements and constraints can usually be identified at the start. Some areas where we believe that performance can be specified are:

1. Growth potential--software may need to be modified to handle future increases in workload.

2. Documentation standards--specific types and quality of documentation should be required.

3. Test and acceptance criteria--how software must perform before it will be accepted must be identified.

4. Maximum computer resources allowable--programs should be required to run in less than specified maximum computer time and capacity.

5. Maintenance--software should be designed and written so that it can be corrected, changed, or modified as simply as possible.

6. Transfer--software should be designed and written to facilitate its transfer from one computer to another.

We found that much of the rework, and many disputes over whether the contractor had performed according to the contract, could have been avoided had these factors been defined at the outset. In our first case, the user requirements that the agency thought were adequate were actually useless. Upon discovering this, the agency allowed the contractor to develop specifications. The lack of adequate user requirements forced the agency to make many changes and create delays as the contract progressed. The absence of testing left the agency with no way to inspect the product.

In our fourth case, the agency allowed the contractor to develop the criteria by which he would be judged; for documentation, the agency merely referred the contractor to agency standards. In our ninth case, the contract required that documentation conform to county standards, which were nonexistent when the contract was signed. The contract also let the contractor choose between two programming languages.

Agencies overcommit themselves too soon, and fail to control contractors through strict phasing

In this context, phasing means stipulating in the contract that work be done in logical phases, with mandatory agency approval of each phase before proceeding to the next phase. When properly used, phasing provides one of the strongest methods of control available to an agency that is contracting out for software developmemt. Phasing provides the agency with the following controls and advantages:

--Initially, phasing helps identify critical milestones which can be used to construct a milestone chart and an overall timetable for the development project. Such a timetable allows the agency to monitor the contractor's progress throughout the contract period.

--Phasing allows the agency to assure itself that the software will be developed in a systematic and orderly manner.

--Phases can be spaced so that the quality and acceptability of the contractor's work is determined before additional funds are spent.

Phasing allows the agency to determine at the end of each phase that

--the contractor is following sound development practices;

--the contractor's work demonstrates a clear understanding of agency requirements;

--the contractor's proposals are technically feasible; and

--the phase under review, and other phases completed to date, represent an adequate base to support the later phases of development.

The agencies that were the subject of our seventh and ninth case studies did not adequately review their contractors' work. In our seventh case, the agency committed itself to software development without a system design, i.e., a description of what the system was to do. A CPA firm which later reviewed the attempt to develop the software said that "the system was never really designed." Since there was no design, it was very difficult later for the agency to withhold payment for poor performance.

In our ninth case, the county committed itself to a firm-fixed-price contract before it had any clear idea of what its system was to do--that is, without functional specifications. The extreme degree of commitment of this type of contract was entirely inappropriate in this situation. It later proved impossible to require the contractor to finish the work for the fixed price. County programmers later spent about $17,000 trying to make the system work and eventually the county was able to use the system.

The completed and approved phases must be left as nearly intact as possible so that the phasing concept is not destroyed. (The effects of excessive changes are discussed further in a following section.) While the case studies indicated that some agencies made at least an effort to phase the contracts, the phasing was not always satisfactorily accomplished. We saw such examples of nonphasing practices as work being done in the programming phase before the design phase was approved and user requirements not fully determined even though work had begun on advanced phases. These examples illustrated the agencies' failure to use phasing as a management technique to monitor, control, and direct contract software development.

Agencies do not manage software development contracts during execution

The case studies showed an excessive number of system changes requested by the agency; the changes ranged from adding minor requirements to changing the entire system concept. In some cases the contractor told the agency that making the changes was troublesome or unnecessary, and in many cases the changes were made well into the contract period. Also, contractors indicated that agency-initiated changes in work scope contributed to the contract's cost and time overruns. In addition, changes undermine the development effort for the following reasons:

--Changes are not usually as thoroughly researched as original design concepts and sometimes have unforeseen effects on other parts of the system.

--As mentioned above, effective use of contract phasing can be destroyed by constantly making changes to work that was completed and approved in earlier phases.

--Under conditions of constant change the agency will find it difficult to determine the exact status of the project at any given time. Consequently, it

may become aware of potential cost and time overruns much later than it would otherwise.

--Excessive changes make it difficult to hold the contractor responsible to perform according to the initial terms of the contract.

A thorough review of each phase of development to ensure that it meets the agency's needs, followed by "freezing" the products of that phase, will allow later phases to proceed with less disruption. Minor changes may be introduced into the system maintenance process with less disruption after the system is implemented. If a certain phase does not meet the agency's needs, it should be immediately reviewed and altered where necessary.

Five of the nine cases we studied did not properly review each phase of development. In our first case, the agency made no attempt to reevaluate its contracting decision when the basic assumptions of it proved wrong. Also, the agency frequently changed the contract's requirements as they were being executed and failed to promptly answer the contractor's questions. Even if the contract should have been continued despite the invalid original assumptions, the frequent changes the agency required as the contract proceeded made it impossible later to deny payment for poor performance. In our sixth case, the agency made no provisions in the contract for monitoring the contractor's work and did not attempt to formally do so. The agency also requested several changes to input and output data formats and grossly underestimated their impact on the work. Since it was not monitoring the work, the agency was not aware of the serious problems until delivery time.

In contrast, our successful fifth case displayed extremely close agency management during contract execution--thorough benchmark tests and technical reviews were conducted throughout the execution at the contractor site, and excellent communications between the contractor and the agency were maintained. As explained in the case write-up (see app. II), a strong project officer and a well-defined problem helped a great deal, but we feel that the close management during execution would have been significant even without these added factors.

Agencies do not adequately inspect and test software

Most software delivered in the case studies was of poor quality. The poor quality was evident in all phases of development. One means of obtaining higher quality software

is for the agency to require the contractor to maintain a quality assurance function to address the following areas:

Working task	Assures that procedures are in effect for subdividing and initiating work, and that someone is assigned the responsibility of getting the work done.
Configuration management	Controls that will insure that system design does not deviate from the base specifications.
Testing	Various levels of tests should be identified and scheduled to insure the quality of the product as development progresses.
Program design	The program design should be reviewed and evaluated before the programming phase begins to prevent the contractor from having to reprogram if the design is changed.
Documentation	Procedures to insure delivery of up-to-date documentation so the agency will know how to run, modify, and maintain the software.

The contractor's plan should be documented and subject to periodic review by the agency throughout the life of the contract. Although in this instance we are speaking about the quality of the software itself, quality assurance must be a part of the agency management process directed toward receiving the quality of software specified in the contract. The military is now using this technique on software contracts. Specific examples of where quality assurance would have been useful are noted in the first and second cases.

Agencies fail to establish a single focal point for the contractor

Establishing a single point within the agency for contact with the contractor will

--shorten communication lines,

--give the contractor one place to go for answers,

--reduce duplication of effort, and

--give one group within the agency an overview of the whole development effort.

The agency officials in this position should be authorized to get answers and deal with problems quickly and effectively as they arise. If a contractor must deal with more than one organization within the agency, these officials should use their authority to ensure that those organizations cooperate fully. This type of interaction will result in a much better coordinated development effort.

In Case 2, we found that (1) in 4 years, several different project officers were assigned to monitor the contract, (2) the contractor dealt with agency officials other than the project officer, and (3) the contractor worked with seven separate agency bureaus. This situation was counter-productive; the need to inform new project officers was continual, and the weakness of agency contract management was underscored.

Miscellaneous causes observed

The following causes also contributed to the case problems we encountered:

--Poor work done by the contractor, such as incorrect programming (first, second, and ninth cases).

--Excessive turnover of both contractor and agency staff (second case).

--Contractor not qualified for the job (ninth case).

--Agency staff not qualified to do their part of the work (second case).

--Documentation of the agency procedures to be automated was not available (fourth case).

How the successful case differed from the others

One notable exception among the case studies was case study number five. This case showed excellent management of the contracted work by a diligent and competent project officer. Excellent contractor selection criteria, functional specifications, monitoring procedures, acceptance testing, and a strong, consistent management effort were all factors in the successful completion of this software contract.

AGENCIES AND CONTRACTORS AGREED ON SOME CAUSES OF PROBLEMS AND WAYS TO IMPROVE CONTRACTING

Contractors and agency representatives who answered our questionnaire often agreed on the causes of problems and on ways to improve software development contracting.

Three of the four problem causes identified by agencies as most important were similarly identified by contractors--(1) requirements were changed well into the contract performance period, (2) acceptance testing procedures were not specified in the contract, and (3) the contract did not stipulate what constitutes satisfactory contractor performance.

Concerning means of improvements, contractors and agency representatives considered the same four actions to be taken by agencies as most important, and ranked them in the same order: (1) define satisfactory contractor performance in contracts, (2) include specific acceptance testing requirements in contracts, (3) specify exact documentation required in contracts, and (4) designate someone who will be authorized to decide on questions which arise.

Both groups also considered the same four actions by contractors as most important, but ranked them differently: (1) request agencies to specify acceptance testing requirements on contracts (ranked highest by both groups), (2) leave the same staff on contracts from start to finish, (3) request agencies to specify documentation requirements on contracts, and (4) work more closely with agencies before contract award to clarify the contract language.

CHAPTER 4

CONCLUSIONS, RECOMMENDATIONS, AND AGENCY COMMENTS

CONCLUSIONS

Federal agencies' contracting for custom computer software development needs improvement. While many software development contracts are successful, too many deliver nothing or deliver something that costs much more and takes much longer than originally estimated. Many Federal software development contracts in the general management category that have experienced trouble share common problems. We believe that the actions we recommend could improve the situation significantly.

Many Federal software development contracts in the general management ADP category have serious problems. At the root of those problems are five important causes. First, a lack of guidance is common to nearly all such contracts. The central agencies--OMB, NBS, and GSA--have not issued adequate guidance on software development contracting, and most other agencies have not attempted to fill that gap by publishing specific guidance of their own. The second cause is the failure of Federal agencies' top management both to realize the difficulty and the importance of software development contracting and to commit appropriate management resources and adequately trained project officers to it. The third cause is the tendency of Federal agencies to commit themselves to software development contracts before user function, testing, and performance criteria have been so defined as to justify their degree of commitment.

Such inappropriate levels of commitment happen because agencies have overly-optimistic assessments of the work they do before contracting and because agencies do not divide contracts into phases so that they can cancel the later phases if earlier ones are not satisfactory. Contractors particularly emphasized that agencies contract without adequate test data. When agencies do not adequately define performance, they have no hope of later denying payment for poor performance.

The fourth cause of contracting problems results when an agency does not manage the work of software development contracts adequately during its execution. Changes to the scope of work, expeditious handling of contractor questions, and inspecting work progress are all handled poorly.

Lastly, serious contracting problems can arise when agencies do not adequately inspect and test software. When a contract originally is inadequately specified and then is not adequately inspected and tested, the agency not only gets bad software but also cannot deny payment to the contractor.

RECOMMENDATIONS

We recommend that the Secretary of Commerce, through the National Bureau of Standards, and the Administrator of the General Services collaborate to issue specific guidance to assist Federal agencies in recognizing and managing the unique factors involved in contracting for custom software development. We recommend that they collaborate because both technical and procurement considerations are involved. The following areas should be covered in the guidelines:

Procurement (primarily GSA's responsibility):

--Internal agency management practices necessary to properly write, manage, and monitor software development contracts in the general management category of ADP.

--Specific instructions on how to tailor software development contracts to the agency's stage of system development at the time the contract is let.

--Contract stipulations to subdivide the software development into phases.

--The review and approval procedures to be followed by the agency at the end of each phase.

--How to apply contract clauses to require as a condition of payment, and to inspect for, both the procurement requirements and the technical specifications (see below).

--How to handle changes in software under development with the least disruption.

--The effective use of contract clauses to ensure the agency's ability to deny payment when the contractor does not perform.

--How to ensure that the contractor follows sound system development practices.

Technical (primarily NBS' responsibility):

--Means of specifying and quantifying software quality, including programming practices standards and adherence to programming language standards. We realize that these areas are still changing, but we believe that a number of common sense guidelines exist 1/ whose widespread publication by NBS would help.

--Means of specifying, defining, and enforcing acceptance testing requirements, and documentation standards.

--Means of defining satisfactory performance by the contractor.

--Guidance on performance specifications to be included in the contract to clarify the quality requirements for the software.

One way that the working-level project manager could get more specific guidance would be for NBS and GSA to work together to design a series of model contracts of various types, including sample clauses covering the withholding of payments, testing, etc. A full explanation of the necessity for, and the advantages of the clauses should accompany these models. The agencies could use these models by extracting the clauses that fit their particular situation and creating a "modular" contract with adequate provisions and safeguards. Although this process would benefit the agency contract procurement function in general, it is primarily recommended as an aid to agency software contract management. Regardless of the methodology used, guidance should not be so broad and general as to exclude specific procedures.

We also recommend that Federal agencies which contract extensively for software development train project managers in appropriate software, contracting, and management skills. These managers should be trained in the GSA/NBS guidance as soon as it is published. Meanwhile, we offer the lessons illustrated in this report and the checklist shown in appendix I to help with contracting for software development.

1/For example, forbidding COBOL programmers to use the ALTER verb because it makes their programs more difficult for others to understand.

AGENCY COMMENTS

We asked the General Services Administration and the Department of Commerce to comment on our draft report. Their replies, which indicated general agreement, are included as appendix III to this report and are discussed below.

The Administrator of General Services agreed with our conclusions and recommendations and stated that the Automated Data and Telecommunications Service of GSA would welcome the opportunity to collaborate with NBS in carrying out the recommendations.

Addressing our recommendation that NBS and GSA collaborate to issue guidance to Federal agencies, the Administrator said that the Automated Data and Telecommunications Service is developing contracting guidance for it own use in providing services to other Federal agencies and that the Service welcomes the opportunity to collaborate with NBS on extending this guidance to other agencies.

Concerning our suggestion that guidance could be provided with model contracts, he said that the Service is also developing standard terms and conditions for software development contracting as part of its contract services program and would like to collaborate with NBS in this effort.

With respect to our recommendation that Federal agencies train project managers, he said that the Automated Data and Telecommunications Service either provides training or lends project managers on a reimbursable basis.

The Department of Commerce generally concurred with our recommendations. Both the Assistant Secretary for Science and Technology and the Department's Office of Procurement and ADP Management provided comments. However, neither made any reference to collaborating with GSA on governmentwide guidance.

Concerning our recommendation that Federal agencies train project managers and ensure that appropriate contracting action is taken, the Commerce Department's Office of Procurement and ADP Management said that it is planning a series of procurement briefings and is distributing our provisional checklist throughout the Department. The Office also indicated that it recently published guidelines for the preparation of technical packages for ADP solicitations. A Commerce official told us that these actions are for use within the Department only.

With regard to our recommendation that NBS collaborate with GSA in developing governmentwide guidelines, the Assistant Commerce Secretary for Science and Technology stated that additional guidelines for the unique problems of contracting for custom software are not in the program plan approved by the Congress. He stated that other NBS products, including programming language standards and documentation guidelines, could be applied to software developed by contractors, as well as to software developed in-house. He also forwarded NBS' specific comments. (See app. III.) Specific comments 1, 2, 4, 5, 6, and 7 were incorporated into this final report almost exactly as NBS sent them.

Concerning the agency's third comment, we changed our wording to follow more closely the wording of an earlier NBS response to our inquiry about the scope of its mission. In response to NBS' eighth comment, we have added a brief proposal evaluation phase to our provisional checklist (See app. I), which includes the elements NBS suggests. However, the added phase was not "developed extensively" as NBS suggests because extensive development is what we are recommending that NBS do--that is why our checklist is called provisional.

APPENDIX I

PROVISIONAL CHECKLIST FOR SOFTWARE DEVELOPMENT CONTRACTING

This checklist, which we prepared during our review, lists matters which we feel agency management should consider when contracting for computer software development. The checklist is divided into five contract phases: pre-contract, contract, proposal evaluation, performance period, and post-contract.

While this checklist is only an interim document, we feel that it will be useful to persons involved in contracting for software development.

We are aware that some of the points included in this list might apply to contracting for any type of service and may be covered in general procurement guidance. We list those points here to make the list complete and to emphasize that they can indeed be applied to software development contracts as well as to others.

APPENDIX I

PHASE: Pre-contract

OBJECTIVES: To obtain realistic assessments of the size, scope, and cost of the effort to produce the needed software.

METHODS:

--Describe the needed software as completely as possible in the request for proposal so that the contractor can understand and address the full scope of work in his proposal.

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

--Develop criteria to evaluate how reasonable and accurate the contractor's costs are.

PROCEDURES:

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

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

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

POSSIBLE WAYS OF ACCOMPLISHING THE PROCEDURES:

--The use of agency jargon, which might be unclear to outsiders, should be avoided.

--A contractor's accounting system must be determined to be adequate to generate valid cost estimates.

--The cost of past development efforts may be compared to current proposed costs.

--Caution should be exercised on bids that are much higher or lower than the average of bids received.

PHASE: Pre-contract (cont.)

OBJECTIVE: To insure the contractor most suited to do the work is selected

METHOD:

Develop comprehensive proposal evaluation standards to use in identifying nonresponsive bidders and in selecting a qualified contractor.

PROCEDURES:

--Consider the contractor's management qualifications.

1. Does the contractor's organization reflect adequate management overall?

2. Is adequate management available for this project?

3. Will the staff responsible for the proposal also work on the project?

4. Is the responsibility for this project at a high enough level of authority in the contractor's organization?

5. Is it likely that key personnel will remain with the project from start to finish?

--Consider the contractor's technical approach.

1. Does it reflect a knowledge of the agency's mission and functions?

2. Will the proposed software take full advantage of the agency's latest computer hardware capability?

3. Does the contractor use software performance measurement tools and techniques as well as software optimization tools to insure the most efficient development possible?

4. Does the technical proposal rely on state of the art or unproven methodology as opposed to proven technology?

5. Is the overall design sound and feasible?

APPENDIX I

--Consider the contractor's quality assurance program.

1. Does the contractor have a documented program in effect that will ensure that the software meets contract specifications?

2. Is contractor quality assurance measurement compatible with the agency acceptance criteria for the final product?

--Consider the contractor's cost proposal.

In doing so, refer to the criteria in the first objective of this phase.

PHASE: Contracting

OBJECTIVE: To initiate a framework for the software development that ensures the maximum chance for success by requiring that sound system development practices be followed.

METHOD:

Include in the contract a requirement that the contractor not bypass system development steps in the development process. This will provide a means for the agency to assure itself that the contractor is performing the required system development steps.

PROCEDURES:

--Determine the system development steps accomplished by the agency to date.

--Identify the remaining system development steps to be accomplished by the contractor.

--Specify the sequence in which these steps are to be accomplished.

POSSIBLE WAY OF ACCOMPLISHING THE PROCEDURES:

As part of the deliverables that the contractor must provide, include a product at the end of each system development step which demonstrates that the step has been satisfactorily completed. These products would include preliminary surveys, feasibility studies, general and detail design, test data and test plan, the actual programs, and acceptance test results.

APPENDIX I

PHASE: Contracting (cont.)

OBJECTIVE: Write a contract which clearly states the obligations of both the agency and the contractor.

METHODS:

--Specify in the contract's statement of work who has the responsibility for each task.

--Specify in the contract what constitutes satisfactory performance by the contractor.

PROCEDURES:

--All agency obligations should be included in the milestone chart and their performance period should be specified the same as the contractor's.

--Tasks should be scheduled so that agency work that is prerequisite to contractor performance will not delay the contractor.

--The contract should provide for phasing the software from the contractor to the agency, including agency training and briefing the staff during the work.

POSSIBLE WAYS OF ACCOMPLISHING THE PROCEDURES:

Whether the specifics of the software are known or not, satisfactory performance should be quantified in terms of all known requirements and constraints. Some areas for performance standards are the

--ability to meet agency's functional requirements,

--growth potential requirements of the system,

--time constraints for deliverables,

--test and acceptance criteria which must be met,

--programming language standards and practices standards to be followed,

--documentation standards to be followed,

--ease of modification, and

--maximum computer resources allowed.

APPENDIX I

PHASE: Contracting (cont.)

OBJECTIVE: Write a contract which gives the agency the right and the visibility to manage the contract.

METHODS:

--Provide for a minimum of investment of funds before the quality of the contractor's work is known.

--Specify who is authorized to make changes in the contract.

--Provide some means to monitor the contractor's progress.

PROCEDURES:

--Phase the development into logical work steps.

--Require agency approval for each phase before beginning work on the next.

--The more undefined the software is, the closer the phasing steps should be at the outset.

POSSIBLE WAYS OF ACCOMPLISHING THE PROCEDURES:

--Besides the contracting officer, specify management's technical representative who is authorized to make changes and to answer contractor's questions.

--Use phasing to assist in setting up a milestone chart showing the time frame for each work step.

--Require the contractor to document the satisfactory completion of each workstep by the agreed date.

--Interim progress reports may be required from the contractor to allow progress to be monitored during each work step.

--Identify performance as well as functional specifications.

--Specify the measures of reliability and quality by which the contractor's work will be evaluated.

APPENDIX I

PHASE: Contracting (cont.)

OBJECTIVE: Write a contract which gives the agency some recourse if the contractor fails to perform.

METHODS:

--The contract should provide for the agency to terminate the contract if it becomes evident that the contractor cannot perform according to the contract's terms.

--The contract should also provide separate due dates and costs for each deliverable and a provision to withhold payment for incomplete or unacceptable work.

PROCEDURE:

If agency evaluation of the contractor's work indicates that the approach being used will not produce the needed software, and if the contractor's work cannot be redirected, the contract should be terminated to minimize losses.

POSSIBLE WAYS OF ACCOMPLISHING THE PROCEDURE:

--Satisfactory performance criteria, and acceptance testing criteria should be used to identify work that does not meet contract requirements.

--Payment to the contractor should be reduced by the amount of any deliverables (e.g., documentation) specified in the contract but not produced.

PHASE: Proposal evaluation

OBJECTIVE: Ensure that a skilled and responsible contractor is selected.

METHOD:

Only contractors with verified technical skill and financial responsibility should be selected to develop custom software contracts.

PROCEDURES:

Agencies should evaluate each proposal in terms of relevant factors, including those listed below, and including considerations of feasibility, comparison to other proposals' prices and schedules, and, if possible, comments solicited from the contractor's prior customers.

POSSIBLE WAYS OF ACCOMPLISHING THE PROCEDURES:

--Include in the request for proposal a provision for agency representatives to visit the contractor site to investigate and evaluate various factors, including finanical position, technical capability, and labor resources.

--Try to determine whether or not the contractor's capabilities, qualifications, and experience are commensurate with the complexity of the problem.

--Try to determine whether or not the contractor's staff has experience with the software and hardware to be used during development.

--Consider the effect on the contract schedule of the contractor's experience (e.g., delay in training the contractor's staff).

PHASE: Performance period

OBJECTIVES: The agency should manage the contract during execution in a manner that contributes to its success.

METHODS:

--The agency should provide all of its deliverables to the contractor within the specified time frames.

--Management should create an environment within the agency that supports the contractor's efforts.

PROCEDURES:

--Such things as studies, requirements, and general designs, to be provided by the agency should be provided promptly so that the contractor is not delayed.

--If provided by the agency, such work products should be complete and accurate and provide a basis for the contractor's work.

POSSIBLE WAYS OF ACCOMPLISHING THE PROCEDURES:

--Management should be involved at high enough levels to insure cooperation by all agency personnel. If there is disagreement about the new development effort among various factions within the agency, it should be resolved in-house by management and not left for the contractor to encounter.

--If possible, the same agency staff should be kept on the project throughout the contract.

--A single focal point should be established within the agency to deal with the contractor.

--An open line of communication should be maintained with the contractor. However, undocumented informal communication can lead to cost overruns. Any changes in the scope of work must be handled by formal modification of the contract document.

APPENDIX I

PHASE: Performance period (cont.)

OBJECTIVE: Monitoring the contractor's progress to ensure that critical milestones are met and to approve work segments.

METHODS:

--Use time frame criteria that are established to determine whether the contractor's development is on schedule.

--All work should be reviewed at the end of each completed phase to determine if it conforms with contract specifications.

PROCEDURES:

--Check progress against dates on the milestone.

--Be alert for signs of problems in the contractor's progress reports.

--Conduct any independent reviews of the contractor's operation that are considered necessary.

POSSIBLE WAYS OF ACCOMPLISHING THE PROCEDURES:

--Consider how well agency requirements are met.

--Decide if the contractor's approach is technically feasible.

--Obtain users' evaluation of contractor's work.

--Render timely management decisions on all alternatives presented by the contractor.

--Once a phase is approved, freeze that phase until development is complete to stabilize the base for succeeding phases.

--Acceptance testing may be applied to completed phases as well as at the end of the development effort.

APPENDIX I

PHASE: Performance period (cont.)

OBJECTIVE: Do adequate testing to ensure that the product meets contract specifications.

METHOD I:

--Make sure acceptance test criteria are still current.

--Maintain adequate control over the testing operation.

PROCEDURE:

If test criteria and data were developed in the beginning, make certain they have been revised to incorporate system changes, if any.

POSSIBLE WAYS OF ACCOMPLISHING THE PROCEDURE:

--Observe or participate in software test.

--Adequately analyze the test results.

--Document all errors revealed by the test.

--Require the contractor to correct all discrepencies as a condition for final payment.

--Follow up on all discrepancies to make sure they are corrected before the system is accepted.

METHOD II:

Assure that personnel responsible for testing the system have adequate technical expertise.

PROCEDURES:

--Assign qualified agency staff with systems, data processing, and performance evaluation expertise to test software.

--If the agency does not have the expertise to adequately evaluate the software, arrange for an independent evaluation by outside sources.

APPENDIX I

PHASE: Performance period (cont.)

OBJECTIVE: To exercise all remedies on behalf of the agency in case the contractor fails to perform.

METHOD:

Full payment should not be made to the contractor until it is determined that the software meets contract specifications.

PROCEDURE:

Exercise the contract recourse provisions discussed under contract controls to the degree that nonperformance was encountered.

APPENDIX I

PHASE: Post-contract

OBJECTIVE: Identify both good and bad aspects of the software procurement.

METHOD:

Make a followup analysis of the software development procurement contract to evaluate contracting practices, record lessons learned, and evaluate user satisfaction with the product.

PROCEDURES:

--Identify practices that are weak and need to be changed.

--Identify and retain practices that produced good results.

--Identify additional agency guidelines that need to be developed and implemented.

--Evaluate user satisfaction with the software.

--Record the actual amount of maintenance programming work that is needed soon after the software is put into production.

--Retain performance data on the individual contractor for future reference.

APPENDIX II

CASE STUDIES IN CONTRACTING FOR COMPUTER SOFTWARE DEVELOPMENT

Following are summaries of the cases we examined in detail during our review. Typically, several causes contributed to the problems in each case. Several of these cases came to our attention because they were known to be problem situations, and therefore, these cases cannot be taken as representative of the majority of Federal software development contracts.

However, we offer these case descriptions to illustrate some of the things that can go wrong--and some of the things that can be done right--because we feel that others may be helped by the experience described.

Case Study 1 - The payroll system that never paid anyone

An agency contracted for the design, programming, testing, and documentation of a payroll system for its headquarters, regions, and service support locations. The cost and time were originally estimated at about $300,000 and 1 year. In the end, the contract cost $1 million and lasted 4 years. Furthermore, the agency found the delivered system unsatisfactory and eventually abandoned efforts to make it work.

What happened during the contract

It appears that the agency made the following assumptions when contracting for the system:

--The user requirements and loose general design developed by the agency would allow the contractor to proceed with the detailed design phase.

--An existing payroll system developed in-house could be adapted to meet agency requirements.

--Based on the above, the payroll system would cost about $300,000 and would take about 1 year to develop.

The contract called for doing the work in three phases: system organization and design; detailed logic design and program specifications; and programming, testing, and documentation. The agency was to approve each phase before the contractor could begin work on the next. Satisfactory performance was defined only in general terms and no provision was made for denying payment for poor performance or lack of performance by the contractor.

Acceptance test procedures were not clearly defined; the contractor was to provide test specifications that would allow the agency to test the system. Although the contractor could suggest changes, only the agency could initiate them. The programming language was specified, and documentation was required to meet standards published by the agency. The agency was to designate one employee to serve as the official representative. No subcontractors were involved.

The contractor's project manager stated that two things became evident early in the contract--the user requirements developed by the agency were useless and modifying the earlier payroll would not meet the agency's needs. As a result, the contractor had to develop a valid set of requirements.

Thus, the assumptions and estimates on which the contract was based became invalid after about 9 months' work: (1) the initial user requirements and system concept were not usable because the system concept changed from decentralized to centralized, (2) modifying the existing system was no longer feasible because it was a small-scale, poorly-organized, second-generation system, (3) because the first two assumptions proved invalid, the original time and cost estimates had to be revised.

Besides the change in the system's concept, the agency made many other changes during the contract. Many of the changes were made to phases already approved, and after work had begun on succeeding phases. The changes were additional requirements and changes in design specifications; one was estimated to have taken about 6 months. The contractor had to ask for a freeze on all system changes several times before the agency finally agreed near the end of the contract.

The contractor stated that he had difficulty getting agency management to make timely decisions on critical issues and that this difficulty contributed to the overrun. At one point the contractor tried to get the agency to agree to a consolidated schedule for the completion of contract actions by both the contractor and the agency. The schedule was never agreed to.

After about 4 years, the agency brought the system in-house to finish and install. In testing it, the agency made several observations. The payroll system did not perform as expected 1/ for the agency's payroll department in retirement, payroll distribution and projection, late and amended time and attendance sheets, or Fair Labor Standards Act requirements. Also, the system could not pay the required number of employees in the 2-week pay cycle.

Besides its deficiencies in user terms, the system had technical deficiencies: (1) it had 18 unexpected failures to continue processing 2/ during the test, (2) it required excessive computer time for processing, (3) it required excessive intervention by programmers to keep it running, (4) the programs in the system contained faulty logic and questionable programming techniques, (5) the documentation was

---

1/In ADP terms, it "did not automate these user functions."

2/Called "crashes" in ADP terms.

not detailed enough, and (6) the old decentralized punched-card-input concepts were perpetuated throughout the system. The agency finally abandoned its efforts to make the system work.

Why this contract failed

The contract omitted important provisions or only mentioned them in general terms--satisfactory performance was not defined, the agency had no recourse in the case of unsatisfactory performance, no acceptance testing procedures were established, and software quality assurance was not required by the contractor.

The agency incorrectly assessed its system development status at the time the contract was let. The overoptimistic, incorrect assessment had these effects:

--When the contract assumptions became invalid, the contractor's scope of work increased, thus increasing the contract's length and cost.

--The attempt to adapt work done under the old concept for use with the new concept caused problems in the new system. The fact that the computer programs still reflected the old concepts contributed strongly to the excessive complexity and unsatisfactory performance of the new system.

--The agency's pre-contract work was wasted because the contractor had to re-do it.

The agency apparently did little to verify that the existing payroll system could be adapted, and we have no indication that it reassessed costs when this proved impossible. The increased work to design a completely new system not only increased overall costs, but may have eliminated the cost advantage which was used to justify the contract.

Besides increasing time and costs, the excessive changes initiated by the agency had secondary effects.

--Much of the advantage of phasing that the contract called for was lost because the agency made changes in previously approved and completed phases (for example, the formerly approved requirements definition was changed after programming started). The changes destroyed the approved work, schedule, and quality control of the earlier phases. Hence, the agency lost much of its ability to control and monitor the direction of the system's development.

APPENDIX II

--The agency complained that the program documentation did not match the programs; the large number of changes undoubtedly contributed to this situation.

The frequent attempts by the contractor to get the agency to "freeze" the system give some indication of the problems the changes caused.

Agency management's failure to make timely decisions certainly contributed to the contract overrun. In an overall schedule, agency management did not list its own areas of responsibility as it had those of the contractor. Management was not strong enough to make the agency's subdivisions follow the overall plan so as not to delay the contract. This internal disagreement not only delayed the contract, but also made it impossible for the agency to hold the contractor accountable for his responsibilities.

Even if the clauses were included in the contract to allow the agency to deny payment for unsatisfactory performance, they would have been useless because the agency contributed to the causes of inadequate contractor performance. In this case study, if the contract had contained such clauses and if the agency had not delayed the contract, the agency apparently would have been justified in withholding payment because of the poor product the contractor delivered and the lack of documentation.

The poor overall quality of the product indicated that the agency should have included a contract provision for a contractor software quality assurance program that would detect, analyze, and correct deficiencies in the software during development.

Case Study 2 - A $2-million, contractor-developed management information system replaced by a $20,000 in-house version

In this case, an agency contracted for a management information system that would allow it to monitor activities in its seven bureaus. The contract called for the Government's liability to be limited to $378,147 and the work to be done in about 9 months. The contract actually cost $2,200,000 and lasted 44 months, during which time it was modified 16 times. Despite the extra time and funds spent, the agency could never use the system and finally abandoned attempts to do so. Agency employees later developed a much less sophisticated system which met the basic requirements and cost about $20,000.

What happened during the contract

In an earlier contract, the contractor had developed a small system for one of the agency bureaus. A contract modification asked the contractor to expand that system to be used agencywide. Under this modification, the contractor identified the objectives and developed a general implementation plan for the larger system. Parts of the first system were to be adapted to the new one. Because of the preliminary work done under the modification to the earlier contract, and because it was believed that part of an existing system could be used, the agency perceived itself to be well into the system development cycle when this contract was let.

The cost-plus-fixed-fee contract was awarded as a sole-source contract because of the contractor's experience and knowledge of the system. The work to be done was based on two documents submitted by the contractor in the original proposal: (1) the conceptual design which specified system requirements and the system description and (2) the implementation plan which outlined the tasks necessary to accomplish detailed design and system development.

The contract was faulty from the outset. The contract specified that inferior and unsatisfactory workmanship would be corrected at no extra charge; however, the contract did not specifically define satisfactory performance or specify acceptance testing procedures. The implementation plan called for monthly progress reports, but the contract did not contain documentation standards; they were prepared by the contractor after the contract was awarded. The statement of work did not require the contractor to provide any evidence of a quality assurance program.

The problems of this contract were many. First, the people who had worked on the system previously developed for the agency no longer worked for the contractor. It was their knowledge and experience that originally prompted the agency to award the contract to the contractor on a sole-source basis. Second, the contractor notified the agency soon after the contract was awarded that the programs developed for the earlier system could not be adapted to the new system.

The agency memorandum recommending award of the contract stated that the agency bureaus had reviewed the conceptual design submitted by the contractor and agreed that it met their requirements. Certain bureau representatives, however, disagreed. It was under these conditions that the contract effort began.

From the outset the contractor experienced difficulty in delivering work on time to meet agency approval. The contractor had originally proposed to complete an agency-approved system design in 6 months. To accomplish this, the contractor had to promptly submit documents describing the system design and operation to the agency for review. After 6 months, the contractor still had not submitted all the documents for approval, and none of those that had been submitted was completely satisfactory to the agency. After 6 more months, the agency approved the system design but continued to ask the contractor to change it. The contractor advised the agency that even small changes could seriously affect the work to be done.

After the design was approved, the contractor's effort shifted to system implementation, which was to be completed in 4 months. After 3 months, the implementation had begun in only two of the seven bureaus. The contractor then proposed to continue implementation work for 16 more months at at an added cost of $700,000. The proposal also expanded the contractor's role in helping the agency control system operations. The agency approved $360,000 of this amount and agreed to develop common definitions, categories, and procedures for assigning descriptive data to like transactions in all seven bureaus. This task proved too technical and difficult for agency staff not familiar with automatic data processing. As a result, about 94 more staff-months of contractor effort were needed.

The contractor stated that system implementation was hindered by (1) agency-requested system changes, (2) difficulties in getting all the bureaus to fully accept and use the system, and (3) system production problems, such as those caused by errors in both software and keypunching. At this point the agency approved an additional contractor proposal

to continue work for a year at a cost of over $900,000. That proposal was supposed to insure more efficient operation and to expand the system to provide additional information. Actually, almost two-thirds of that cost was for the contractor to continue work he was already doing rather than for improvements.

The system was installed on the computer at the agency's data processing center. The agency found that the system was too complex, that it required too much data preparation, and that management could not use its reports. The contractor proposed to simplify the system and eliminate some reports, and the agency approved additional work for these purposes.

An agency official stated that both poor contractor performance and agency failings contributed to the contractor's cost overrun. He stated that the contractor's project manager did not use professional system development practices and project control techniques. For example, programming started before system design was approved. The official added that the contractor did not submit progress reports and the agency did not conduct periodic technical reviews.

At the end of the contract period the simplified version of the system was incomplete. The agency refused to award funds to continue work on the system and instead, hired two other firms to make the system operational. The new contractors reported that:

--Computer programs were in various stages of completion and ranged from barely adequate to useless.

--Program testing was in worse shape than reported and, in their opinion, if any testing had been done, no one had bothered to look at the results.

--Programming techniques, documentation, and library control all suffered serious deficiencies.

--To bring the programs to a stage where an integrated systems test could be performed would require a great deal of work.

The agency eventually abandoned the use of the original system and developed an in-house system for about $20,000 which basically performed the same functions as the proposed simplified version of the original system.

We reviewed the system development documentation and observed that:

--The agency assigned six different project officers during the life of the contract.

--The contractor assigned at least three different development teams to the project.

--The contract was modified 16 times.

--The programming language that the programs were to be written in was apparently not clearly specified initially. The agency later instructed the contractor to use COBOL, but programs were delivered in PL-1.

--Agreements were made between the contractor and agency officials other than the project officer.

Why this contract failed

The agency's assessment of its system development status when it let the contract was overoptimistic. Also, the contract omitted important provisions, or mentioned them only in general terms. For instance, satisfactory performance was not defined, the agency had no recourse for nonperformance by the contractor, acceptance testing procedures were not established, quality control procedures were not required of the contractor, and no documentation standards were included in the contract. Thus, the agency could not maintain control over the type and quality of the software being developed.

The agency assumed that existing software could be adapted to the new system and that contractor personnel who were knowledgeable of agency operations would be available. These assumptions proved false. The agency's incorrect assumptions--showing an incorrect assessment of its stage of system development--led to two situations harmful to the agency's interest.

First, the contract was awarded sole-source without competition, and second, the cost and time estimates were based on what the agency thought had been accomplished and the amount of work thought to be remaining. If the agency had determined that it was not technically feasible to adapt software from the old system to the new system and that the contractor's supposed experience would thus be lost or compromised, better estimates might have shown that the system was not feasible to develop.

None of the project officers the agency assigned to work with the contractor proved a strong focal point for the control and resolution of problems that came up during the

contract. To compound this situation, the contractor had to deal with seven bureaus which, according to the contractor, offered less than their full cooperation. Also, the large number of agency project officers and contractor teams assigned to the project made communications very difficult and a successful contract almost impossible.

The agency lost another form of control because of the lack of strict phasing of the project. In this case, the contractor began work in the programming phase before requirements were fully determined; programming started before the system design was approved, and the requirements and users were changed frequently during the contract.

The complexity of the delivered system and the excessive data preparation work it required indicate that the agency had done little if any technical review of the contractor's design work. They also indicate a lack of performance specifications and minimal system acceptance testing.

The poor quality of the product delivered by the contractor was exhibited by serious deficiencies in the programming technique and documentation. Those deficiencies indicate that the agency should have required the contractor to demonstrate that quality control procedures were in effect during the software development effort. They also indicated that the agency should have specified acceptance testing requirements.

In summary, the agency failed to (1) negotiate a contract which identified and required the specific level of performance expected of the contractor, (2) recognize the complexity of the system when the extensive contractor-generated specifications were presented to them, (3) determine what would be needed to support, use, and maintain such a system, (4) establish a knowledgeable project officer as a single focal point to deal with the contractor and to give that person the necessary authority to ensure the cooperation of all organizations within within the agency, (5) aggressively monitor the contractor's progress and pay for only the agreed to levels of performance, (6) accomplish the system development tasks originally identified as their responsibility, and (7) insist that the work be done in distinct, sequential phases with adequate technical review. The result was a large, complex system which cost many times the original estimate and was useless.

Without a doubt the contractor's work was inadequate. However, the agency should have written and managed the contract to ensure that it had some recourse if the delivered product proved to be useless.

Case Study 3 - The payroll/personnel system that wasted $1 million

The agency let a contract to design an integrated personnel/payroll system, redesign an expense accounting system, prepare an equipment requirements statement, and develop a background study on all its ADP systems. The original contract cost and time were $445,158 and 15 months, most of which was for the software development portion of the contract. The agency finally terminated the contract after 28 months and $970,000 and with no usable software being delivered.

What happened during the contract

When it issued a request for proposal, the agency had only identified a need for the software and was still in the initial stages of system development. It had not fully developed user requirements or system specifications for any of the proposed software.

The agency selected a contractor and awarded a fixed-price contract. The contract required that the software development be phased but did not require agency approval of a completed phase before work began on the next phase. Procedures were written to control changes in the scope of the work, and the programming language was specified. The contract only described satisfactory performance by the contractor in such general terms as "the system must demonstrate effectiveness, be automated wherever possible, and be flexible." The contract did not contain acceptance testing procedures and the documentation the contractor was to produce was identified in the contract, but the quality criteria for it were not.

The contract's delivery dates, scope of work, and costs were revised several times, yet the contractor was still unable to meet the revised schedule or deliver an acceptable product to the agency. The contractor stated that he was unable to meet the delivery schedule due to extensive changes requested by the agency and inexcusable delays caused by the agency. Agency officials admitted that some of the changes requested were not clearly identified in the contract and that others were clearly outside the scope of work.

The contractor maintained that the agency took too much time to review products submitted for approval. The agency admitted that the reviews were not timely but maintained that the length of the review was due to the poor quality of the documentation to be reviewed.

The contractor did not clearly understand the software systems the agency desired because the contract did not specify system requirements or performance criteria. Both agency and contractor staff agreed that the contract was not specific, that the terminology was vague, and that many system requirements were not clearly identified.

The contract included a milestone chart showing starting and completion dates for phases of system development. The contractor requested the agency's approval after phases were completed but did not wait for that approval before starting work on succeeding phases. For example, the contractor began work on the detailed system design before receiving agency approval of the general system design. Later, the agency rejected the general design and required the contractor to make several changes, so work already done on the detailed design had to be scrapped or reworked.

User requirements were never adequately defined and frozen. The agency provided system concepts for two of the systems, and the contractor worked with current system users, reviewed available system documentation, and developed user requirements based on responses to a questionnaire. The contractor used this information to develop the general design which the agency rejected several times for not meeting requirements. Each time the agency rejected the design, it added new requirements. The changes delayed completion schedules, increased contract costs, and caused the agency and the contractor to disagree about whether the new requirements were included in the original scope of work.

The contract was amended 13 times to provide for additional work to be done, to add or delete requirements, and to reimburse the contractor for extra costs due to delays caused by the agency. The amendments were to increase the cost of the contract to $1,037,448. The agency eventually became convinced that the contractor could not deliver at an acceptable time and cost, cancelled the contract, and tried to withhold payment for poor performance. A negotiated settlement price of $970,000 was finally agreed to. None of the software was ever used by the agency.

Why this contract failed

The fixed-price contract was not suitable since the agency could not provide detailed specifications for the software desired. Because the contract did not include detailed specifications, it had to be modified 13 times to accommodate changes, extend delivery dates, and provide more money. One form of cost-plus contract is normally better for cases where the exact scope of work has not been identified. In this case,

APPENDIX II

however, the contractor wanted to limit his efforts to make a profit at the fixed price he bid. The relationship between the agency and the contractor deteriorated as requirements were added, delays incurred, and the contractor saw his profit shrink.

Phasing was not used to good advantage in this contract. The system development was not broken into fixed phases with agency approval required for each before proceeding. As it was, even the requirements were still under dispute when the contract ended. The parties could not use the contract to settle disputes because it was written in such general terms.

The areas in which those generalities caused significant disputes, delays, and added costs were acceptance testing, performance specifications, and documentation.

In summary, the agency wrote a contract which was (1) the wrong type for the stage of system development they were in when it was let and (2) too general to give the contractor a clear understanding of what was to be done.

Also, the agency did not use phasing to control the development process and managed the contract so poorly that the contractor could show that the agency contributed to the failure to produce usable software and was thus able to defeat the agency's attempt to withhold payment.

Case Study 4 - The failure to develop a centralized accounting system

The agency, which used nonuniform accounting systems, contracted for the design and development of a centralized accounting system to increase responsiveness, timeliness, accuracy, and to overcome inefficiencies in the operation of 10 accounting offices within the agency. The cost and time spent were estimated to be $958,682.40 and 27 months. After 30 months the system was only about one-fourth complete, and the agency cancelled the contract. Although the system was not complete, the agency paid about $981,200.

What happened during the contract

At the time the contract was let, the agency had no formal design or specification documents for the contractor to work from. The agency had collected a list of concepts and standards which supposedly were the basis of a conceptual design. The cost-plus-fixed-fee contract obligated the contractor to deliver a workable accounting system.

The contract called for three development phases. Each covered the development of a major accounting subsystem, with the first phase to include the overall system design. Each phase was further divided into conceptual design, detail design, and implementation. The agency was to approve each phase at completion. Under the terms of the contract, the contractor was to develop a project control plan for such items as progress and cost reporting, documentation review, and acceptance testing. The contractor was responsible for formulating the criteria by which the agency would judge his performance. The contract called for the contractor to submit proposed changes along with the reasons for them to the agency for approval. The agency reserved the right to make modifications it considered necessary to ensure that the system fit its needs. System documentation requirements were fairly detailed, but guidance on program documentation only referred the contractor to agency standards. No subcontractors were involved.

In the first phase, the contractor was to develop a general design of the overall system and also the design for one major subsystem. In the development of this phase the contractor stated that he encountered two problems--(1) the agency staff generally resisted the new system and (2) virtually none of the existing accounting processes and procedures, which the new system was to automate, was documented. When he submitted his report on the first phase, the contractor assumed he could immediately start on the next phase, but agency review of the report took about 250 staff-days. The

contractor said that agency delays and the low level of agency participation together added about 350 staff-days.

As the contractor entered the second part of the first phase, he still encountered problems he blamed on (1) the poor quality of agency review and agency staff participation, (2) agency indecision, and (3) agency changes in direction. The contractor felt the changes deviated from earlier agreements and that some of them could not be made. Some products were submitted for agency approval three or four times. Disagreements arose over the amount of documentation necessary, and the lack of existing agency procedures continued to be a problem.

The contractor contended that the agency insisted on a system that was not needs-oriented but one which was designed to satisfy many individual preferences. To illustrate his point, he compared the excessive number of management reports asked for in this system (188) to the maximum number of reports (44) called for in four other agencies' accounting systems. The agency director of systems admitted the general lack of direction and specifics in the contract and stated that more definitive planning and guidance were needed to let the contractor know what was expected.

To determine what, where, and how data would be stored and retrieved, the contractor asked the agency to specify (1) the computer hardware to be used, (2) the data base management system (DBMS) 1/ to be used, and (3) system requirements, such as output reports and transaction coding. The agency took about 6 months to provide guidance in these areas, and during that time, the contractor proceeded to design conventional file structures and processing routines.

When the agency finally decided on the DBMS, the design had to be substantially reworked. The contractor said that even small changes generated extensive reviews to determine all other areas which were affected and required change. Of the six major reasons given by the contractor for overruns, three dealt with changes that were constantly being made to both the requirements and the operational environment and the impact those changes had on system development.

1/A Data Base Management System (DBMS) is a computer software package which can facilitate the management, manipulation, and control of data.

Agency officials maintained that the contractor's report reflected a general lack of understanding of the job to be done, that deliverables were inadequate, and that agency documentation standards were not followed. Conversely, the contractor stated that agency staff assigned to work on the subject did not understand the agency, its mission, or its needs.

After about 2-1/2 years, the agency had paid the contractor over $981,000. The system was estimated to be only about one-fourth complete, and the time frame had exceeded the original estimate by several months. At this point the agency terminated the contract. The contracting officer said that the agency's counsel had informally advised him that a precisely defined set of requirements was never incorporated in the contract. Therefore, since neither party could define the product, in their unofficial opinion, the agency could not force the contractor to finish the system for the maximum cost allowed by the contract.

Why this contract failed

Too many factors were left to be subjectively determined outside the provisions of the contract. This condition is evidenced by such things as arbitrary changes, disagreements on various subjects, and the agency's admissions that more specific requirements should have been given to the contractor. These problems can be avoided even if the exact characteristics of the needed software are not known at the time the contract is let.

First, user requirements, performance specifications, quality control procedures, acceptance testing procedures, and documentation items required can be specified in the contract to establish a framework at the outset. Second, the agency should have required the overall system design to be defined in a first phase to the point that its adequacy could have been determined, approved, and frozen to allow stable and systematic development before committing itself to the rest of the contract.

Other factors which contributed to the failure of the contract included the agency's failure to:

--Establish firm, realistic requirements and fix them.

--Render timely decisions and timely review of products.

--Promptly carry out responsibilities so that software development was not delayed and so that the way was

left clear to invoke contract penalties against the contractor if he failed to perform.

--Maintain adequate monitoring and tracking procedures which would have avoided allowing the entire original contract amount to be spent in the first of three development phases.

--Define the user requirements served by the existing accounting system.

--Create an environment which enhanced chances for success, including consensus on agency needs, proper orientation of agency staff, and provision of a strong focal point of qualified agency staff to work with the contractor.

Case Study 5 - The successful compiler development

In this case the agency contracted for the development, implementation, and documentation of a SIMSCRIPT II language compiler for the agency's computers. The contract called for the compiler to be completed and delivered in 1 year at a cost of $119,800. The software was delivered within the cost and within a month of the delivery date with final testing to be accomplished shortly afterward.

What happened during the contract

The agency had developed detailed specifications. Also, the agency had developed proposal evaluation criteria, which included acceptance testing criteria. The fixed-price contract specified the system requirements as well as what was expected of the contractor. It included a 26-page statement of work with several appendexes detailing the required and desirable system specifications.

The agency's acceptance of the product was to be based on the software's successful completion of a final comprehensive benchmark; however, interim tests were to be conducted throughout the performance period. The contractor was obligated to correct any deficiencies found in the product for up to 3 months after delivery at no additional cost. The contract cited the exact documentation expected and also provided for the contractor to train agency personnel to use the compiler. Contract provisions required that the contractor's principal staff on the job have experience in developing and implementing large-scale simulation language compilers, and that they have at least 2 year's extensive use of the SIMSCRIPT language. The contractor's progress was to be monitored by on-site reviews at the contractor's facilities.

To begin with, the agency developed good proposal evaluation criteria. The project officer stated that the criteria were developed to provide a way to reject vendors who would probably be unable to do the job. Second, he wanted to evaluate the proposals in terms of their dollar benefits, considering the future savings to be generated by each rather than just their development cost. Finally, he wanted to be able to differentiate between the qualified vendors. This project officer was deeply involved throughout the proposal evaluation and prepared the majority of the specifications and requirements.

To accomplish these aims, the project officer devised a 41-step weighted rating system to rate the proposals in three areas: the vendor's compliance with mandatory contract requirements; the vendor's capabilities, qualifications, and

experience; and the vendor's predicted product performance and dollar benefits to the agency. He also inserted several clauses in the contract that he knew were technically infeasible, to trap ignorant vendors.

The contract also contained a clause allowing agency representatives to visit the vendor's facility to determine his ability to perform. This visit, if conducted, could investigate and evaluate current financial position; technical, production, and financial capability; plant facilities and equipment; quality assurance; labor resources; performance record; and other appropriate subjects.

The project officer was also concerned with the quality of the functional system specifications. The procurement officer said they were the best he had seen for a software procurement; the clear specifications allowed him to offer the contract on a competitive basis. The contractor stated the specifications made it very clear what the agency wanted.

The software development incorporated benchmark tests which were conducted throughout the development cycle. A benchmark monitor was designated by the agency to develop the benchmark programs, conduct the tests, and prepare the test results. The agency specified the exact documentation the contractor was to provide, including user manuals, systems programmers' manuals, complete flow charts, and complete source code to maintain the compiler after the contract expired and to facilitate competitive contract maintenance. Training was provided by the contractor to allow agency perpersonnel to conduct followon systems maintenance. The contractor delivered all documentation and training to the agency's satisfaction.

With his experience in ADP in general, and in simulation languages in particular, the project officer was equipped to monitor the contractor's progress. To track and evaluate the contractor, he incorporated the benchmark testing process into the contract and conducted periodic on-site reviews with contractor and agency staff.

Each on-site progress review lasted 2 or 3 days. According to the project officer's trip reports, at these meetings he was apparently able to determine how well the contractor was progressing, to discuss and resolve problems, and to generally assure himself that the contractor was adequately fulfilling his contractual obligations. Further assessment of contractor progress was provided by the periodic benchmark tests. Also, throughout the project's development, contractor and agency personnel communicated openly and frequently to assess contractor progress and to resolve questions that arose.

Delivery of the software was delayed 1 month because the contractor could not obtain documentation from the hardware vendor for the execution library. The agency considered this delay unavoidable and it was mutually agreed upon. After some minor problems were corrected, the software passed the final benchmark test on September 15, 1972, and final payment was made to the contractor.

According to the agency, it received an excellent product. The compiler was usable immediately after acceptance and saved the agency an estimated $65,000 annually because of its increased efficiency. Also, the contractor delivered the documentation according to plan.

Why this contract succeeded

This case is an example of successful software development contracting. Several very important factors were in its favor:

--The agency had an extremely diligent and thorough project officer who was extremely competent and technically knowledgeable. This was verified by a contractor official and the agency's contracting officer, who said that the project officer's functional specifications made it possible to have a competitive procurement. Moreover, this project officer stayed with the project from start to finish.

--An extensive contractor rating and proposal evaluation scheme was developed which ensured that only capable contractors would be selected.

--Excellent functional specifications were developed including extensive benchmark tests (both during development and during acceptance), documentation, and training.

--Periodic reviews and open communication were maintained throughout the contract.

--A type of computer programming task was included which could definitely be made a fixed-contract requirement. That is, the compiler was to translate a given programming language (SIMSCRIPT II) into machine language for a specific brand of computer. The language did not change during the contract. Also, the performance and efficiency of the compiler could be more easily quantified to provide benchmark test criteria. The relative importance of having a programming task less changeable than the typical business programming task is difficult to assess. However, we believe that it is important.

APPENDIX II

Case Study 6 - The contract test scoring program that failed its own test and had to be rewritten by agency programmers

The agency contracted for the development and implementation of a data storage, retrieval, and editing system to be used in connection with a certification test program the agency conducts. The tasks to be accomplished consisted of three major data processing programs, a test data sheet, and 19 additional auxiliary computer programs. The programs were to make the testing easier and to automate much of the later analysis and retrieval of test results. No detailed description of these programs was included in the contract. Originally the contract was supposed to be completed in 6 months at a cost of about $14,000. The contract lasted about 9 months without usable software being delivered. Even though no usable software was delivered, the agency still paid the contractor about $13,000.

What happened during the contract

Agency officials stated that when the contract was let, they had identified "fairly specific" system requirements and had developed a general system design. They wanted their existing program to be drastically modified to make it more efficient and to provide more data. The agency apparently perceived itself to be well into the system development cycle when it let the contract.

The contract was written generally without detailed task descriptions. The agency relied on verbal discussions with the contractor to define requirements. Provisions were made in the contract for:

--All code to be written in Fortran IV, internally documented (commented) 1/ enough that an equally skilled programmer could easily modify and enhance the programs.

--The contractor to provide, whenever possible, input/data base file formats and layouts, outputs layouts, and system flow diagrams on interrelationships between data files, data flow, program utilization, and file usage.

1/ Comments in FORTRAN are notes for persons to read that are embedded among the actual FORTRAN computer language statements of the program.

The contract did not provide any procedures by which the agency could monitor the contractor's performance. In fact, it stated that written progress reports from the contractor would not be necessary. The contract did not establish any procedures which would have facilitated monitoring, such as phasing, defining specific benchmarks and acceptance testing procedures, establishing procedures to control changes in the work scope, and specifying detailed documentation standards.

The contract does not clearly describe what work was to be performed or what products were to be delivered. Agency officials later stated that they had a verbal agreement with the contractor and that both parties clearly understood the anticipated system design, the system specifications, the user requirements, and the software application. Further, agency officials said it was clear that they expected the contractor to deliver finalized programs in time for implementation at a specified point. The agency said the contractor was involved in early discussions of the task and had agreed with the agency's estimated number of hours required to complete the work. The contractor stated that the agency had not developed details of the task and that he was engaged in part to help formulate these details.

Although only two official modifications were made to the contract, we found evidence that several informal changes were made during the performance period. The two formal modifications did not affect the substance of the work to be performed; they designated a new agency project officer and extended the contract completion date.

The informal changes were more substantial. The agency requested several changes to input and output data formats within the next several months after the contract was let. The agency and the contractor disagreed on their significance. Agency officials believed that the changes were not substantial and cited the fact that the contractor never indicated that the changes were causing him delays or inconvenience. The contractor, however, stated that the changes did cause delays.

Agency officials underestimated the complexity and the amount of programming support needed to complete one program. They estimated 192 hours, but contractor staff actually spent more than 350 hours on it and estimated that 220 more hours would be required.

The agency relied on informal monitoring and did not require formal progress reports from the contractor. The agency project officer, who was the contractor's primary

contact with the agency, stated that he relied on the contractor's verbal reports to monitor progress. Later he said that he felt the contractor staff wasted time when they were not watched.

The agency did not receive usable, completed software. When the contractor could not successfully demonstrate the product about a month before the scheduled completion, the agency realized it would not be finished by the end of the contract period. None of the computer programs finally delivered was suitable. The first program was inefficient, costing between $250 and $300 per day to run and over $2 to generate a single test score. It contained many programming errors and was not adequately documented. The agency never even tried to use the second program the contractor delivered. Agency programmers eventually wrote a program which cost about 90 percent less to run than the first program delivered. The agency withheld a final payment of $1,351.81 from the contractor because of the inadequacy of the software delivered.

Why this contract failed

The agency failed to correctly assess the system development progress it had made before letting the contract. While agency officials stated that they had identified requirements and completed the general system design stage, both the generalities of the contract documents and the contractor's testimony made it clear that this was not true. Since the agency had not completed a system design and did not ask the contractor to, development work was begun without a foundation of design. Also, without a design, criteria for acceptance testing could not be developed even if the agency had attempted to include it in the contract.

The vaguely written contract did not make the contractor's responsibilities clear. Verbal discussion was substituted for formal contract requirements, and the lack of communication between the agency and the contractor never allowed the scope of work to be clarified.

The agency made no provision for monitoring the contractor's progress or for reviewing and approving work phases. Consequently, the agency was not aware of problems until it was nearly time for the product to be delivered. Furthermore, the agency defined the work to be done so vaguely that little basis existed on which interim performance could be evaluated. No formal progress reports from the contractor were required, and changes in the scope of work were handled informally by the agency. According to the contractor, these changes had a great impact on the project and caused the original completion date to be expanded.

In summary, the agency wrote a contract which (1) did not clearly define and document what was expected from the contractors, (2) did not phase development with review and approval of each phase, (3) did not provide for the contractor's progress to be adequately monitored during the course of the project, and (4) was based on a stage of system development not yet reached.

APPENDIX II

Case Study 7 - The three-party contract that failed to produce a usable accounting system

In this case, a Federal agency contracted the administration of one of its functions to an association. The association is responsible for maintaining accounting and administrative records of the function's operations. The association then contracted with a computer firm to develop and implement a computerized accounting system. For purposes of this analysis, the three parties will be referred to as the agency, the association, and the contractor.

Under the accounting system designed and implemented by the contractor, the association was unable to prepare financial statements and to properly identify the function's operating costs. A CPA firm reviewed the association's records and concluded that if an audit could be performed at all, it would have to be so extensive that the cost would make it prohibitive. Considering system and contractual problems to be insurmountable, the association cancelled the contract and assumed control of the system. At this point the association had paid about $4.3 million of the $5.7 million billed by the contractor.

What happened during the contract

The association used system specifications prepared under a previous contract to solicit bids for the development of the accounting system. Even though the specifications were used as a basis for the solicitation, they were later estimated to be only from 50 percent to 80 percent complete.

The total contract cost an estimated $16 million over 5 years. However, for purposes of this study we used only the development, implementation, and conversion costs. Those costs were to total $1,308,252 with work exceeding the scope of the contract to be quoted on the basis of time and materials used based on rates listed in the contract. The contract called for the work to be done in four stages but did not stipulate that each stage would be completed and approved by the association before work could begin on the next. Indeed, the milestone chart indicated that work might be done concurrently on all stages.

The contract required the contractor to draft an acceptance test plan for each stage of development. The contract stipulated that the agency could withhold payment for each calendar day's delay. The contractor was also liable for damages sustained by the association because of negligence or failure to perform unless it was caused by the association.

APPENDIX II

Written authorization from the association was required before the contractor could make modifications. The programming language was specified in the contract, and the documentation desired was described. However, quality standards levied on the contractor were stated in general terms, such as, "the highest quality standards reasonably obtainable in the data processing industry," and "the system will edit data to insure validity," without any measures of either quality or validity.

The contract was let in August 1974. Stage III was completed in May 1975. The association's methods and procedures committee reviewed and approved the acceptance test plan for each stage.

In November of 1974, the committee approved Stage I of the system development. In their approval letter they stated that Stage I programming appeared to be reasonably sound and many portions of Stage II programming were already well underway.

Despite the testing and approval of each stage, numerous problems were encountered immediately after the system was placed in operation and throughout 1975. The number of problems incurred makes it appear that the quality of the associations' review of the work in progress was poor. The association later concluded that reports generated by the system were grossly incorrect.

The problems became so severe that eventually the association was unable to prepare financial statements on the results of its operations. The association's controller attributed this inability to multiple financial and statistical problems caused by the failure of the computer contractor to meet his contractual obligations. The association requested a CPA firm to review the problems experienced with the contract. The firm reported that the computer system did not meet the criteria established by the association for accountability of program operations and cited about 75 deficiencies covering 22 major system requirements.

In March 1976, the association made an offer to the contractor to pay about 70 percent of its unpaid billings, which had been withheld to settle the contract. The contractor demanded formal arbitration. On November 1, 1976, the arbitration proceedings were terminated by mutual consent of the parties. The termination agreement was formally executed containing a settlement of $4.1 million on the contractor's claim of $5.7 million. Meanwhile the association had paid the CPA firm over $1 million in fees to review the system, provide support for arbitration, test, and conduct system planning. The association assumed control of the system once the contract was terminated.

APPENDIX II

According to an analysis of the system by the association and by the CPA firm, an estimated 6,900 more staff-days would be needed to bring the system up to an acceptable level, as a result of the system deficiencies mentioned above. The association manager stated that the system design had never been completed and that the contractor had deviated from the fragments of design that did exist.

In a report on the condition of the system, the agency's Inspector General listed the following deficiencies:

--Numerous controls outside the system have had to be established to assure the integrity of the data processed.

--The system does not provide the needed statistical data.

--Very little documentation (i.e., systems manuals, procedures, etc.) was obtained from the contractor.

--The existing system lacks the standardization normally associated with integrated systems, (e.g., inconsistency in program terminology).

--The system is not designed to facilitate modifications to meet changing requirements.

The report also stated that the system did not perform some of its planned functions and performed others incorrectly or inadequately.

The contractor pointed to instances where association management took excessive amounts of time to render decisions necessary to resolve matters and thus delayed efficient construction and operation of the system. He also claimed that the data provided by the association as system input contained errors.

The agency's Inspector General concluded that adequate requirements had not been included in the initial system design and that the specifications approved later proved to be inadequate. Also, neither the association nor the agency provided the technical expertise needed to adequately monitor the contractor's development of the system.

The association did not create an environment conducive to successful contracting, and agency management failed to recognize and deal with the following factors:

APPENDIX II

--All conditions were not addressed in the contract and/or specifications.

--Delays occurred because of untimely management decisions.

--Provisions were not made to insure the quality of input data.

Failure to properly manage the software contract resulted in:

--The loss of financial control of the agency function.

--Expenditures of over $1 million for systems analysis and arbitration support from outside consultants.

--The additional cost to bring the system up to acceptable levels of performance.

Why this contract failed

Two factors jeopardized the success of this software contract from the start. First the system specifications were incomplete when the contract was let and they were never completed by anyone. Second, the contractor's work was reviewed poorly or not at all.

When the CPA firm reviewed the system it concluded that the first step necessary to bring the system up to par was to complete the specifications "because the system was never really designed." At best, the specifications were only 80 percent complete. Also, the specifications implied too many things rather than being specific about any of them. The incomplete system specifications indicated that the agency's system development posture was not as advanced as it was perceived to be.

The poor quality of association and agency review of the contractor's work was first reflected in the acceptance of incomplete specifications from the previous contract. Then, in the present contract, a system which was approved by the association at the completion of each developed stage showed serious problems immediately after installation.

APPENDIX II

Case Study 8 - The engineering technical information system that was developed from inadequate technical information

In this case, the contract was for the development of a major subsystem of a large automated system designed to maintain and retrieve engineering data and related information to support the agency's engineering and procurement function. The subsystem is designed to provide data to aid engineers in reconstructing the original configuration of a major part or hardware assembly before modifications were introduced. The original cost and time of the subsystem contract were to be $93,039 and 28 months. Contract modifications and cost overruns increased the final cost to $123,726 in 28 months.

What happened during the contract

The contractor was to develop a formalized plan along with detailed procedures for implementation as well as computer programs needed to establish the subsystem. The subsystem was to be developed as a modification to an earlier contract. The subsystem contract did not define standards or criteria for measuring product quality. Timetables were provided within the contract for completion of individual requirements. Although no standard test data package existed for acceptance testing, the contractor was required to demonstrate the ability of the computer programs to operate and to correct or replace any unsatisfactory work.

The software printed three reports. Agency officials stated that requirements were inadequately defined for one of the reports. A contract modification was required at a cost of $7,902 to correct this deficiency. About 18 months into the contract period, a second modification was negotiated which cost $19,044. Of this amount, about $10,000 was for correcting a deficiency and the rest for adding another function. An agency programmer stated that before the outputs could be used he had to write auxiliary computer programs whose output supplemented the information shown on the contractually-required reports.

Generally the contractor did not meet specified completion dates. The agency delayed payments to the contractor for failure to meet the schedule. About $3,741 was awarded to the contractor for cost overruns, bringing the total paid over the original cost to $21,643, excluding the cost of added functions.

System documentation seemed to be adequate, but the computer programs themselves were not properly documented.

Why this contract failed

The agency did not perform, or contract for, adequate system analysis work. This is indicated by the fact that specifications were not adequately defined to assure that the software would have the necessary capabilities. Quality assurance and testing procedures in the contract were inadequate to assure that the software would meet user needs without modification by agency programmers. Documentation standards should have made program documentation mandatory.

APPENDIX II

Case Study 9 - The court case retrieval system

A Federal agency provided funds to allow a county to develop computer software to automate the criminal court calendar, speed the scheduling of hearings, and to provide current case and defendant information to Federal, State, county, and local agencies. The county let a fixed-fee contract costing $74,405 for the development of this software. The contract was to last 12 months but actually took about 36 months. County programmers later did about $16,000 worth of extra work, and the programs were eventually used.

What happened during the contract

The county did not do a significant amount of development work before contracting, so the contractor had to establish system requirements--one of the initial steps in software development. No contract provision existed to allow the county to review the contractor's work in this initial step and cancel the rest of the contract if not satisfied. The contract required that all specifications and documentation conform to the county's data processing standards; however, county programmers said that no written program standards existed when the contract was signed. The county programmers did say that unwritten standards were discussed with the contractor, but because the contractor did all the coding off county premises, the county could not review the code before delivery to ensure compliance.

The contract included timetables for completion of individual requirements; however, work under the contract and subsequent closing agreement was about 3 years later than originally negotiated. The original contract provided that the county would approve a test and approve the adequacy of program documentation before making final payment.

The closing agreement attached a time limit for acceptance by the county. It required that the contractor programs be tested and evaluated no later than 2 months after submission. Also, the closing agreement required that if items were found to be unacceptable, the contractor would make any corrections or modifications quickly. If any items were not corrected by the contractor 2 months after the date of first receipt of all items, the county was to pay only that proportion of the remaining moneys due for those items that were accepted. Full payment was not to be made until all items were tested and accepted.

The contract contained a "termination or curtailment" clause which stated that if the contractor failed to perform any of the requirements, the county could terminate the agreement.

Also, the contract required the contractor to:

--Submit a report analyzing requirements and general recommendations to improve or revise the existing criminal court procedures.

--Develop a detailed system to satisfy the above report.

--Generate specifications and documentation for use by the agency's data processing office.

The report was to include existing procedures for handling the flow of information through the court. Also, flowcharts of existing methods of processing cases and defendants and an analysis of forms and documents used by the county were required. The contractor was also to recommend a new system concept, including descriptions, flowcharts, new and revised forms, and a general approach to a data-processing-oriented system of information retrieval for the county.

The contract listed numerous items which were to be included in or delivered with the systems specifications, the programming requirements, and the documentation. The systems specification items included a system requirements document which was to specify production and user delivery dates, workload, and controls; description of major computer and clerical processes; and narrative showing the flow of documents, hand and machine processing, and documentation of identified user needs. The programming requirements included adherence to county standards for programming language (a choice of two was allowed), and the delivery of decks and listings. The documentation requirements list included functional flowcharts, program flowcharts, narrative descriptions, file layouts, and a data element dictionary.

However, despite the numerous items named in the lists, the contract contained nothing specific describing how the items were to look, and the county standards cited did not exist on paper.

County officials stated that the contractor's first system proposal was disapproved because it appeared that the contractor merely mechanized the existing manual system instead of streamlining procedures to exploit the computer. The contractor submitted a revised system proposal which the county approved.

One county official stated that the contractor claimed criminal court system expertise that he did not have. Another official cited examples of the contractor's poor

APPENDIX II

understanding of court procedures. Contractor representatives acknowledged that they had demonstrated a "less than perfect performance."

About 2 years from the start of the contract, the parties entered into a "closing agreement." The contractor agreed to complete work on specified contract items while the county agreed to complete all other contract requirements plus those additional items not found in the initial contract. The contractor was released from the requirement to program monthly and annual reports. A county official estimated that it would cost the county an additional $15,000 to $18,000 to write the programs themselves.

Almost 3 years after the originally negotiated completion date, the contractor completed and the county accepted all items under the terms of the closing agreement.

County officials said that most of the contractor's computer programs--about 60 percent of the program statement--had to be rewritten by the county. Programs also required extensive error correction 1/ before they would work properly. The software documentation did not include program logic flowcharts or sufficient explanation of complex processing logic.

Why this contract failed

The contract itself lacked specificity. Documentation, which later proved to be a problem, was required to make the system conform to data processing standards of the county, but the county had no written program standards when the contract was let. Also, since the contractor's first proposal showed no understanding of the county's mission, the county apparently did not furnish the contractor with adequate functional specifications. Another indication of the lack of specificity is that the contractor was allowed to use either of two programming languages.

The county also had minimal monitoring, review, and quality assurance procedures in effect. This was illustrated by the slipped milestone dates, the fact that much of the software had to be redone, and the in-house cost of from $15,000 to $18,000 to do work which was originally the contractor's responsibility. These are all effects of poor contractor performance and poor county contract management.

1/ called debugging

General
Services
Administration Washington, DC 20405

JUL 11 1979

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

Dear Mr. Staats:

We have reviewed the General Accounting Office draft report entitled "Contracting for Computer Software Development -- Serious Problems Require Management Attention to Avoid Wasting Additional Millions," dated May 24, 1979, and agree with its conclusion and recommendations.

The following comments are offered in response to specific recommendations:

1. ". . . that NBS and GSA collaborate to issue specific guidance to assist Federal agencies . . ." (page 28). As part of its Contract Services Program our Automated Data and Telecommunications Service (ADTS) is developing contracting guidelines for its own use in providing software development services to other Federal agencies. We welcome the opportunity to collaborate with NBS on extending the benefits of these guidelines to other agencies.

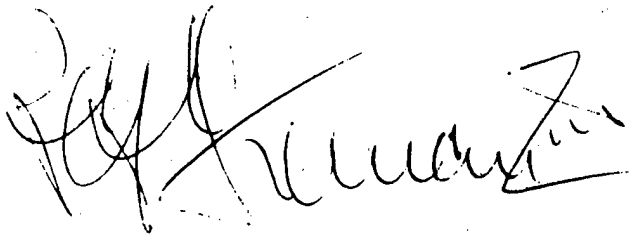
2. "One method to provide more specific guidance to the working level project manager would be for NBS and GSA together to design a series of model contracts . . ." (page 29). ADTS is also developing Standard Terms and Conditions for software development contracting as part of its Contract Services Program, and would like to collaborate with NBS in this effort.

3. ". . . that Federal agencies which contract extensively for software development train project managers in the pertinent software, contract and management skills needed" (page 29). In its Manpower Services Program, ADTS provides software development project managers to other Federal agencies on a reimbursable basis. As in the two areas mentioned above, we welcome the chance to share our experience with other agencies, by providing on-the-job or formal training, or both.

APPENDIX III

Our Automated Data Telecommunications Service (ADTS) will be available to work with the National Bureau of Standards to implement the report's recommendations.

Sincerely



R. G. Freeman III
Administrator

GAO note: The page numbers have been changed to correspond to the pages in the final report.

APPENDIX III

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

July 18, 1979

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

Dear Mr. Eschwege:

This is in reply to your letter of May 31, 1979 requesting Comments on the draft report entitled "Contracting for Computer Software Development -- Serious Problems Require Management Attention to Avoid Wasting Additional Millions". The Assistant Secretary for Science and Technology provided the enclosed comments.

In addition, the Office of Procurement and ADP Management concurs with the recommendations that Federal agencies involved in software contracting (1) train project managers in pertinent skills necessary to manage large software development contracts and (2) ensure that appropriate action is taken in each phase of contracting for software development. They have identified several ways in which the Office of Procurement and ADP Management, as the Department's central procurement activity, can help ensure proper and effective contracting for software development. First, they are planning a series of procurement briefings to be conducted at each Department operating unit to better educate project managers on ADP procurement regulations and techniques. Secondly, they are disseminating throughout the Department the provisional checklist for software development contracting which was provided as Appendix I to the GAO report. Third, they recently published and distributed guidelines for the preparation of ADP solicitation technical packages.

We have reviewed the comments and believe they are responsive to the matters discussed in the report.

Mary P. Bass
Inspector General

Enclosure

APPENDIX III

UNITED STATES DEPARTMENT OF COMMERCE
The Assistant Secretary for Science and Technology
Washington, D.C. 20230 (202) 377-3111

JUL 3 1979

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

Dear Mr. Eschwege:

Thank you for the opportunity to review the draft GAO report entitled "Contracting For Computer Software Development -- Serious Problems Require Management Attention To Avoid Wasting Additional Millions." We generally agree with the scope of the problem outlined in the draft report, and share GAO's concern.

Many of the program products in the Congressionally-approved Federal ADP standards plan for NBS are applicable to problems identified by this report, especially those problems characteristic of the software development process itself. Products such as documentation guidelines, language standards, cost estimation guidelines, programming guidelines, and testing guidelines can all be applied to software developed under contract for the Federal Government, as well as to in-house development and, in many cases, procurement of software products. Additional guidelines applicable to the unique problems of contracting for custom software are not currently in the Congressionally-approved program plan.

The NBS staff has prepared the following specific comments which we hope will be of use to you as the final report is prepared.

1. Page 5, middle: In the first sentence under Software development, add "requirements analysis, system design or specification, and" after the word "into." The corrected sentence should read: "The life cycle of computer software is divided into requirements analysis, system design or specification, and development and operation."

2. Page 6, middle: The sentence at the top of page six and the sentence at the middle of page six are incomplete. It appears that some material is missing from the text.

APPENDIX III

3. Page 15, top: The responsibilities cited in the first paragraph are inconsistent with the responsibilities assigned to NBS under the Brooks Bill (PL 89-306) in Section 111 (f). We recommend that: (1) the words "and guidelines" be inserted after the word "standards" in the fourth line from the top of the page, and (2) the remainder of the sentence and the paragraph be deleted. The revised sentence would read: "The National Bureau of Standards (NBS) is assigned the task of developing technical standards and guidelines."

4. Page 15, bottom: Delete the sentence "NBS representatives recently informed us that their proposed new budget includes a project on software development contracting."

5. Page 37, middle: Add the following statement as the third item under PROCEDURES. "The contract should provide for adequate phase over of software from the contractor to the agency, including such items as agency training and walk-throughs."

6. Page 39, middle: Delete the second item under METHODS and insert the following statement. "The contract should also provide separate due dates and costs for each deliverable and a provision to withhold payment for incomplete or unacceptable work."

7. Page 41, bottom: Add the following statements to the fourth item under POSSIBLE WAYS OF ACCOMPLISHING THE PROCEDURES. "However, undocumented informal communication can lead to cost overruns. Any changes in the scope of work must be handled by formal modification of the contract document."

8. General Comment: Appendix I does not adequately provide for the process of evaluating proposals of software contractors as a separate phase. Suggested items to be used in evaluations should be developed extensively and should include the following elements:

   - Is the expertise of the contractor commensurate with the complexity of the problem?
   - Does the contractor have experience with the software and hardware to be used during development or will training (possibly at Government expense) be required?

APPENDIX III

- What effect on contract start-up time does the contractor's experience have?

This separate phase should be added after page 39 which covers CONTRACTING and before page 41 which covers PERFORMANCE PERIOD.

We believe that this report will prove valuable to the Federal community in improving its contracting practices with respect to computer software development.

Sincerely,

Francis H. Holek

Jordan J. Baruch

GAO note: The page numbers have been changed to correspond to the pages in the final report.

(913550)

U. S. GOVERNMENT PRINTING OFFICE : 1979 620-386/361